Thesis for the Degree of Doctor of Philosophy

# Stochastic Modeling for Visual Object Tracking and Online Learning:

## manifolds and particle filters

Zulfiqar Hasan Khan

Department of Signals and Systems
Signal Processing Group
Chalmers University of Technology
Gothenburg, Sweden 2012

**Stochastic Modeling for Visual Object Tracking and Online Learning: manifolds and particle filters**

Zulfiqar Hasan Khan

ISBN 978-91-7385-641-6



Doktorsavhandlingar vid Chalmers Tekniska Högskola
Ny Serie nr 3322
ISSN 0346-718X

Department of Signals and Systems
Signal Processing Group
Chalmers University of Technology
SE-412 96 Gothenburg
Sweden
Phone: +46 (0)31 772 8060
Email: zulfiqak@chalmers.se
zulfiqar.hasan@gmail.com

Prepared using LaTeX.
Printed by Chalmers Reproservice
Gothenburg, Sweden, 2012

To Soni and Shaheer

**Abstract**

Classical visual object tracking techniques provide effective methods when parameters of the underlying process lie in a vector space. However, various parameter spaces commonly occurring in visual tracking violate this assumption. This thesis is an attempt to investigate robust visual object tracking and online learning methods for parameter spaces having vector or manifold structures.

For vector spaces, two different methods are proposed for video tracking. The first builds upon anisotropic mean-shift tracker for appearance similarity and SIR particle filter for tracking of the bounding box. The anisotropic mean shift is derived for a partitioned rectangular bounding box and several partition prototypes with adaptive learning strategy of reference object distributions. The joint scheme maintains the merits of both methods, using a small number of particles ($<20$) and stabilizes trajectories of target during partial occlusions and background clutter. The second object-tracking algorithm uses joint point feature correspondences and object appearance similarity and an optimal selection criterion for the final tracking. The point feature-based tracking simultaneously exploits and dynamically maintains two separate sets of point feature correspondences in the foreground and surrounding background, where background features are used for the indication of occlusions. The appearance-based tracking uses an enhanced anisotropic mean shift with a fully tunable (5 degrees of freedom) bounding box. The enhancement is achieved by partially guiding it from feature point tracker and dynamically updating the reference object models. It is shown that proposed tracker has more tracking robustness with reduced tracking drift.

The contribution related to manifold tracking and online learning focuses on *symmetric manifolds* (set of covariance matrices) and *Grassmann manifolds* (set of subspaces). The online appearance learning is based on Bayesian estimation of state variables (related to object appearances) on these manifolds by a dual dynamic model. This model is realized through the help of two mapping (exponential and logarithmic) functions between tangent planes and manifolds. The tracking part is based on Bayesian estimation of state variables (related to affine object bounding box) with manifold appearance embedded. Tracking and online learning is performed in an alternative fashion to mitigate the tracking drift. Moreover, for symmetric manifolds, Gabor features in different frequencies and orientations are introduced for the covariance descriptor. This is effective for both visual and infrared video objects. Further, the spatial information is incorporated in the covariance descriptor by extracting features in partitioned bounding box. For Grassmann manifolds, a novel method is introduced to detect the partial occlusion. The appearance subspace is updated in each time interval if there is an indication of stable and reliable tracking without background interferences. It is further shown that the proposed manifold framework is better by comparisons with existing trackers.

**Keywords:** Visual object tracking, anisotropic mean shift, particle filters, Bayesian tracking, consensus point feature correspondences, online learning of reference object, Riemannian manifold, covariance tracking, Gabor features, Grassmann manifold.

# Acknowlegments

This thesis is the result of four and half years of work. There had been ups and down but fortunately many people were on my side when I needed their help and support.

First of all I would like to thank my supervisor Prof. Irene Y.H. Gu for her encouragement throughout my Ph.D. research, for believing in me and giving me the freedom to pursue the research directions I was interested in. She always kept me motivated to analyze new ideas not only experimentally, but also in a mathematically rigorous way. Without her support and invaluable guidance on research related issues, this thesis would have never come into existence. Furthermore, I have to thank Prof. Mats Viberg, for letting me join this research group and for providing rich, creative environment to work.

I am grateful to NESCOM (National Engineering and Scientific Commission), Pakistan and Chalmers University of Technology, Sweden for providing me with the scholarship to conduct this research. I owe many thanks to my colleagues in the signal processing group for their help at different time and different aspects during my study. Special thanks go to Andrew Backhouse, Tiesheng Wang and Peter Strandmark, who worked with me closely and helped me to understand many issues in tracking. I would also like to thank Agneta Kinnander and Natasha Adler for their help with many practical issues. I would like to acknowledge Lars Börjesson for providing excellent IT support.

Of course, life outside the lab did also exist. I greet those who I have encountered along this period and with whom I have shared very special moments. I am grateful to our mini-family of Pakistani friends specially Shahid, Faisal and Saeed. You are just terrific!

Finally, I would like to express my deepest appreciation to my family for encouraging me in my PhD work– thank you mother, brother Mansoor, sister Wajiha and my family on wife's side. Special thanks and love goes to my wife Soni and son Shaheer. Their smiles, love and support gave my life, a meaning and encouragement to achieve career goal.

**Zulfiqar Hasan Khan, Göteborg, 2012**

# List of Publications

The thesis is based on the following publications

**Paper A**

Z. H. Khan, I. Y. H. Gu. and A. Backhouse, Robust Visual Object Tracking Using Multi-Mode Anisotropic Mean Shift and Particle Filters. IEEE Transactions on Circuits and Systems for Video Technology, vol. 21, no. 1, pp. 74-87, March 2011.

**Paper B**

Z. Khan and I. Y. H. Gu, Joint Feature Correspondences and Appearance Similarity for Robust Visual Object Tracking. IEEE Transactions on Information Forensics and Security, vol. 5, no. 3, pp. 591-606, September 2010.

**Paper C**

Z. Khan and I. Y. H. Gu, Bayesian Framework-based Dual Model for Online Learning and Object Tracking on Riemannian Manifold. Submitted to IEEE Transactions on Image Processing, December 2011.

**Paper D**

Z. Khan and I. Y. H. Gu, Nonlinear Dynamic Model for Visual Object Tracking on Grassmann Manifolds with Partial Occlusion Handling. Submitted to IEEE Transactions on Systems, Man, and Cybernetics, Part A, January 2012.

Other publications by this author, but omitted in the thesis

- Z. H. Khan and I. Y. H. Gu, Tracking Visual and Infrared Objects using Joint Riemannian Manifold Appearance and Affine Shape Modeling . Proc. $11^{th}$ IEEE Workshop on Visual Surveillance 2011, Barcelona, Spain, November 13, 2011.

- Z. H. Khan and I. Y. H. Gu, Bayesian Online Learning on the Riemannian Manifold using A Dual Model with Applications to Video Object Tracking. Proc. $1^{st}$ IEEE Workshop on Information Theory in Computer Vision and Pattern Recognition 2011, Barcelona, Spain, November 13, 2011.

- Z. H. Khan and I. Y. H. Gu, Visual Tracking and Dynamic Learning on the Grassmann Manifold with Inference from a Bayesian Framework and State Space Models. Proc. IEEE International Conference on Image Processing (ICIP) 2011, Brussels, Belgium, September 11, 2011.

- Z. H. Khan and I. Y. H. Gu, Adaptive appearance learning for visual object tracking. Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2011), pp. 1413-1416 , May 22, 2011.

- I. Y. H. Gu and Z. H. Khan , Online Learning and Robust Visual Tracking using Local Features and Global Appearances of Video Objects. Object Tracking, Hanna Goszczynska (Ed.), ISBN: 978-953-307-360-6, InTech, 2011.

- Z. H. Khan, I. Y. H. Gu and A. Backhouse, A Robust Particle Filter-Based Method for Tracking Single Visual Object Through Complex Scenes Using Dynamical Object Shape and Appearance Similarity. Journal of Signal Processing Systems, Springer New York, October 2010.

- Z. H. Khan, I. Y. H. Gu and A. Backhouse, Joint anisotropic mean shift and consensus point feature correspondences for object tracking in video. Proc. IEEE International Conference on Multimedia and Expo (ICME) 2009, pp. 1270-1273, June 28, 2009.

- Z. H. Khan, I. Y. H. Gu and A. Backhouse, Joint particle filters and multi-mode anisotropic mean shift for robust tracking of video objects with partitioned areas. Proc. IEEE International Conference on Image Processing (ICIP) 2009, pp. 4077-4080, November 7, 2009.

- T. Wang, I. Y. H. Gu and Z. H. Khan, Adaptive particle filters for visual object tracking using joint PCA appearance model and consensus point correspondences. Proc. IEEE International Conference on Multimedia and Expo (ICME) 2009, pp. 1370-1373, June 28, 2009.

- A. Backhouse, Z. H. Khan and I. Y. H. Gu, Robust Object Tracking Using Particle Filters and Multi-region Mean Shift. Proc. Lecture Notes in Computer Science, 2009, vol. 5879/2009, pp. 393-403, 2009.

# Contents

# Part I
# Introduction

CHAPTER 1

# Introduction

Visual object tracking has attracted a great deal of interest in recent years, largely driven by their applications such as video surveillance in airports, schools, banks, human-computer interaction, hand-gesture recognition, video compression, medical imaging in hospitals, e-health care and many more. Building a visual tracking system is far from being an easy task due to several aspects e.g. most objects are not rigid and can deform freely; occlusions can make objects temporarily invisible; background clutters can make objects boundaries unclear and difficult to distinguish; illumination variations may change object appearance and many more. Additionally, depending on a visual tracking domain, detection of interesting objects and analysis of object tracks to recognize their behavior can be the necessary steps before and after visual tracking. In the last two decades, extensive research has been carried out in this area, and various methods for visual tracking have been proposed, however, many problems in visual trackers are still open research issues.

Informally, visual tracking consists of a recursive estimation of the unknown parameters (often referred to as states) of an object described by a region in an image plane as it moves around the scene. To achieve visual tracking, choosing a target model by utilizing the shape and appearance of an object is one important aspect. The second important aspect is the formulation to find a target model in successive video frames. The mathematical framework for visual tracking has been very well developed (see [2, 31] and references therein). [1] have described two approaches namely bottom-up approach (also called target representation and localization) and top-down approach (also called filtering and data association) for tracking. The top-down approach uses the target dynamics, prior information, control theory principles and probabilistic search to find targets, where they are expected to be. Both these approaches specify a target model, however, they primarily differ from each other in search methodology. The bottom-up approach utilizes the computer vision principles or deterministic search to estimate tracking parameters. In the following, we describe and discuss each of these components separately. Readers are referred to [2] and

references therein for more details.

## 1.1 Target Modeling

The selection of an object shape, appearance features and their proper representation are important aspects in target modelling. [2] describes various ways to represent an object shape. Objects that occupy small regions in an image can be represented by points [3, 4], that is, the centroid. Rigid or non-rigid objects can be crudely represented by a rectangular or ellipse bounding box [1, 5] or more accurately by object silhouette and contours [6], skeletal models [7] and articulated shape models. Using a bounding box is usual for generic algorithms. Different shapes representation can be visualized by Fig. 1.1. This selection is often application-dependent and has

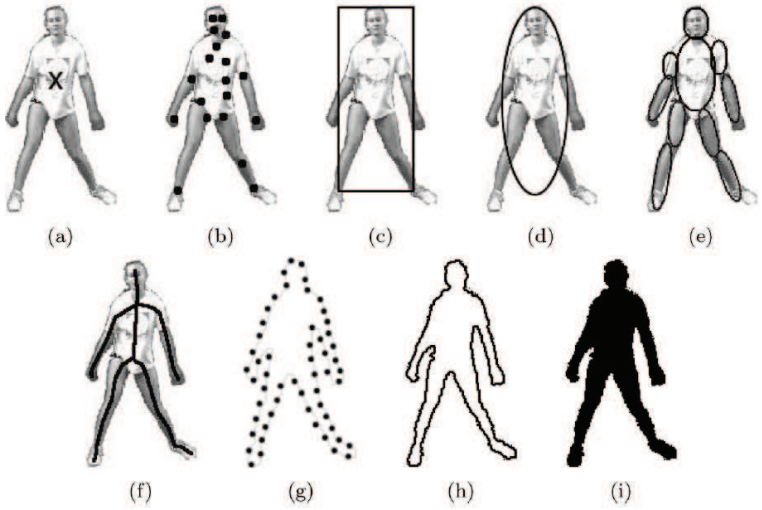


Figure 1.1: Object shape representation. (a) point (b) multiple points (c) rectangular box (d) elliptical box (e) part-based articulated object (f) object skeleton (g-h) object contour (i) object silhouette. The figure is taken from [2].

a critical role for object motion modelling. For example, only translation models can be used if an object is represented by a point; parametric motion models like affine or projective transformations are more appropriate for geometric shape representations like ellipse or square.

Commonly used appearance features in the context of visual tracking are colors (e.g RGB, HSV), edges (e.g. Canny edge detectors, SIFT features), filter outputs (e.g. 1st/2nd derivatives, Gabor filters), optical flows, textures and many more. Appearance features representation may use global properties of an image region specified by shape models e.g. probability densities (parametric or non-parametric) of features [1, 8], templates [9], eigen images [5], features covariance matrix [10] and many more. It may also use local properties of group of neighboring pixels, which

should be unique and distinguish objects from others e.g. Harris corners [11], KLT features [12], SIFT features [13] and SURF features [14] and many more

## 1.2 Target Model Search

Informally, target model search can be categorized into following:

### 1.2.1 Deterministic Methods

These methods define correspondences error function and carries out a target model search in the current frame starting at a previous location of object. The aim is to find the most similar object location that minimizes the error function like sum of square distance, Bhattacharyya distance. The success of these methods is highly dependent on the discriminating power of object representation and the requirement that the object in consecutive frames has a certain overlap. Some examples of deterministic visual tracking methods are template based tracker [15], histogram based tracker [1].

### 1.2.2 Detection Methods

These methods detects targets in every frame by utilizing computer vision principles and then corresponds objects across frames to generate the tracks. Some examples of object detection methods in the context of visual tracking are local feature correspondence [11, 13, 16], segmentation [17, 18], background subtraction [19–21] and supervised classifiers [22–24].

### 1.2.3 State-Space Methods

These methods consider uncertainty in previous frame information and use state-space models and Bayesian filtering to find the target model. The general state-space equations for Bayesian filtering can be described as:

$$\begin{aligned} \mathbf{s}_t &= f_{t-1}(\mathbf{s}_{t-1}, \mathbf{v}_{t-1}) \\ \mathbf{z}_t &= h_t(\mathbf{s}_t, \mathbf{w}_t) \end{aligned} \tag{1.1}$$

In the following, we describe and discuss each of these components separately:

#### State Vector $\mathbf{s}_t$

It represents all information known about the target and may include the shape/appearance of the object or other parameters that are required to describe the complete property of the target.

#### State Dynamic/Motion Model $p(\mathbf{s}_t|\mathbf{s}_{t-1})$

It predicts the next state at time $t$ conditioned on the state in all previous frames or only previous state under the first-order-Markov assumption. The uncertainties in prediction are controlled by the noise variance in each state variable. Commonly

used motion models are Brownian, second-order constant velocity (CV), third-order constant acceleration (CA) (chapter 6 [25]) and many more.

**Measurement/Sensor Model $p(\mathbf{z}_t|\mathbf{s}_t)$**

It describes the relation between sensory readings, under the influence of the surrounding environment and object's state. Measurements usually consist of image region specified by the object state.

**Bayesian Filtering**

The Bayesian filtering gives a recursive method of calculating posterior density $p(\mathbf{s}_t|\mathbf{z}_{1:t})$ as a function of the measurement model $p(\mathbf{z}_t|\mathbf{s}_t)$, motion model $p(\mathbf{s}_t|\mathbf{s}_{t-1})$ and knowledge of prior $p(\mathbf{s}_0)$ at the initial step or previous step $p(\mathbf{s}_{t-1}|\mathbf{z}_{1:t-1})$. It consists of two steps: prediction and update. In the prediction step, motion model is used to find the current state conditioned on previous measurements. In the update step, measurements are utilized to improve the prediction.

There are many ways of formulating the Bayesian filtering under different assumptions. Commonly used techniques are Kalman filters [26], for linear additive Gaussian noise for motion and measurement model, extended Kalman filters [27] and unscented Kalman filters [64] for non-linear motion and measurement models, particle filters [29, 30] for non-linear and non-Gaussian models etc. Moreover, concepts of multiple model filtering and data association are required to solve the tracking problem in case of different modes of object operation or multiple objects scenarios. However, these are beyond the scope of this thesis and not described here.

## 1.3 Online Learning of Target Appearance

Online learning for estimating time-evolving stochastic processes is an important research issue in signal processing and computer vision. One of the main tasks for online learning is to estimate current statistics, parameters or states of a non-stationary system or object from new observations. In the context of, visual tracking that does not have the opportunity of offline training, online learning means that the object appearance model should adapt over time in such a way that it is robust to the object intrinsic parameters (e.g. pose variation, shape deformation) and resilient to the extrinsic (e.g. illumination, camera motion, viewpoint and occlusion) variations using some previous tracked frames. Depending on the assumption for image space, the task of online learning for visual tracking can further be subdivided in two categories, detailed as follows:

### 1.3.1 Appearance Learning on Vector Space

These methods model the appearance of an object in a vector space (i.e a space closed under vector addition and multiplication by scalar) e.g. Images presented as intensity (or color) values, histograms, point features and many more. These spaces are dealt with the machinery of linear algebra or differential calculus. Recently, many object

tracking methods have been proposed and developed [2, 31], which consider the image as a vector space structure. e.g. [32] extends gradient-based optical flow to handel the appearance variations. [33] handles the fast illumination changes by fast differential EMD tracking. [34] introduces an online subspace learning method with a sample mean update. [35] proposes to label objects and background pixels by ensembles of online, learned weak classifiers.

### 1.3.2 Appearance Learning on Manifolds

These methods model the appearance of an object on a manifold that is locally Euclidean. For example, images presented as the set of covariance matrices (the symmetric manifold) or set of subspaces (the Grassmann manifold) are not vector space structure. Concepts of differential geometry and differential calculus are used for the study of these structures. Images of an object may reside in non-linear space [36]. Thus, using manifold techniques for appearance learning and tracking may lead to more robust results. For example, [37] uses a conjugate gradient and Newton's method for subspace tracking on Grassmann and Stiefel manifolds with applications to orthogonal procrustes. [38] proposes piecewise geodesics on complex Grassmann manifolds using projection matrices for subspace tracking. The simulations are performed on synthetic signals from an array of sensors. [39] proposes a visual tracking system by applying a Kalman filter to velocity vectors in the tangent planes of Grassmann manifolds.

## 1.4 Problem Definition and Motivation

Online learning of object appearances enables a tracker to adaptively utilize a timely reference object model. A main challenge for online learning of visual object is to decide whether the ambiguity on image changes should be learned or avoided. The ambiguity can be caused either by object itself (e.g. deformation, pose, self-occlusion) or by other occluding object/background. It is desirable that an online learning method adapts to changes in object intrinsic parameters (e.g. pose, appearance and shape) and insensitive to extrinsic variations (e.g. illumination, occlusion, background, camera motion and viewpoint). Another difficulty is that objects appearance do not always reside in a vector space e.g. symmetric positive definite covariance matrices, subspaces and many more. Moreover, apart from learning, the tracking method should be made robust. For example, by choosing complementary methods that can work cooperatively and benefit from each other and many more.

This thesis is an attempt to address these issues by investigating new approaches in vector spaces and manifolds for visual object tracking and online learning. It is concerned with the objects, captured by a single static or dynamic camera, with the assumption that detection in the first frame is already solved or need not be solved for a setup.

## 1.5 Contributions

The scientific contributions of this thesis can be summarized as follows:

**Tracking method 1: Joint multi-mode anisotropic mean shift and particle filter**

Equations for the multi-mode version of the anisotropic mean shift are derived by partitioning the bounding box into smaller subregions. It is then embedded into the particle filter framework along with online learning of the reference object distribution. This allows mean shift to search several modes within the box. Moreover, a method is proposed to measure the suitability of subregion kernel bandwidth estimate for box shape parameters calculation (i.e. position, width, height and orientation). The new hybrid tracker performs better as compared to mean shift (MS) tracker in [40] and the combined mean shift and particle filter (MSPF) tracker in [41].

**Tracking method 2: Joint object appearance similarity and local point feature correspondences**

A point feature-based tracker is proposed that simultaneously exploits and dynamically maintains two separate sets of point feature correspondences in the foreground and in the surrounding background for visual tracking. It is integrated with appearance-based tracker by an optimal selection criterion. The appearance based tracker utilizes enhanced anisotropic mean shift with a fully tunable (5 degrees of freedom) bounding box and online learning of the reference object distribution. The joint tracker performs better as compared to anisotropic mean shift tracking in [40] and SIFT tracking in [13] followed by the RANSAC [42].

**Tracking method 3: Adaptive tracking on Grassmann Manifolds**

A Bayesian framework is formulated on Grassmann manifolds for the posterior appearance estimation and object bounding box tracking. The online learning uses dynamic model that includes both manifold object appearance point and its velocity. A criterion is proposed to find stable tracking results for appearance learning. The proposed tracker performs better as compared to subspace tracking on Grassmann manifolds in [39] and vector subspace-learning based tracking in [43].

**Tracking method 4: Adaptive tracking on Symmetric Manifolds**

A spatial dependent covariance tracker is proposed on symmetric manifolds that uses Gabor features in different frequencies and orientations on partition subregions. It integrates the tracking process and online learning process in an alternative fashion. The online learning is based on Bayesian estimate of object appearance on a manifold by a dual model. The tracking part is based on Bayesian estimation of object affine bounding box parameters with manifold appearance embedded. Experiments on both visual and infrared videos have shown robust tracking performance as compared to

covariance-based tracking in [44] and probabilistic tracking on the Riemannian manifold in [45].

## 1.6 Thesis Outline

The thesis is divided into two parts. The first part introduces the background and methods. The second part includes publications resulted from this thesis. A brief introduction to the content in each chapter of the first part of the thesis is given below.

**Chapter 2** Riemannian Manifold

This chapter is devoted to describing manifolds in a very informal way. First, concepts of geodesics, exponential map, logarithmic map on manifolds are introduced followed by their descriptions for Lie Group, Grassmann and symmetric manifolds. This theory is aimed at providing essential information required for adaptive visual tracking on manifolds in Paper C and Paper D.

**Chapter 3** Related Methods of Visual Object Tracking

This chapter describes existing methods involved in the tracker design of this thesis. These techniques include mean shift for global appearance model-based tracking, particle filter for state space tracking and SIFT in [13] followed by the RANSAC [42] for local point feature-based tracking.

**Chapter 4** Adaptive Tracking Methods

This chapter summarizes proposed object appearance learning on a vector space with hybrid trackers combining mean shift with local point feature correspondences and Bayesian tracking. Further, it describes the proposed visual tracking and online learning on Grassmann and symmetric manifolds. These algorithms are detailed in attached Papers.

**Chapter 5** Conclusion and Future Work

This chapter contains the conclusion of this thesis. The discussion for future work is also given.

CHAPTER 2

# Manifold Theory: A Review

The study of feature analysis, estimation and optimization on manifolds appears in a vide variety of computational problems in computer vision. Intuitively, manifolds can be thought of as smooth, curved nonlinear surfaces that are not vector spaces. Notions such as sums and differences of points in these spaces are not defined. The purpose of this chapter is to give a general introduction to the theory of manifolds. The fundamental concepts regarding manifold structures, tangent spaces, exponential and log mapping are presented first. Then a discussion of manifolds relevant to this thesis is given. The approach and definitions mainly follow [37, 38, 46–50]. Note that the points on a manifold are represented by small bold letters and matrices are represented by capital bold letters.

## 2.1 Manifolds

Informally, a manifold can be considered as a space that consists of open sets of Euclidean space $\mathbb{R}^d$, glued together and made differential. It can be visualized [51] by considering a circle. Globally, the circle can not look like an open subset of $\mathbb{R}$ because if we travel in a "straight line", then we will end up where we started. This is in contrary to a straight line in $\mathbb{R}$, where we never come back to where we started. However, the circle can be divided in four hemispheres top, bottom, left and right where the angles of all points between 0 and $\pi$ can be thought as open interval in $\mathbb{R}$. The union of these four "open sets of $\mathbb{R}$" can form a circle. Thus, a manifold is locally similar to Euclidean space, whereas globally it doesn't.

Formally, a $d$-dimensional manifold $\mathcal{M}$ is a Hausdorff, topological space that is locally homeomorphic to Euclidean space. For every point $\mathbf{p} \in \mathcal{M}$, there exists a neighborhood $\mathcal{U} \in \mathcal{M}$ containing $\mathbf{p}$ and an associated mapping $\phi$ from $\mathcal{U}$ to some Euclidean space $\mathbb{R}^d$, such that $\phi(\mathcal{U})$ is an open set in $\mathbb{R}^d$ (for a more formal definition, see [50, 52] Chap.1)). There are myriad constructions and definitions associated with topological space, Hausdorff and homeomorphism. For the sake of completion, these

are summarized below (Readers are referred to [JM2006] for more details):

**Topological space:** It consists of a set $\mathbf{X}$ together with a collection of subsets of $\mathbf{X}$, called open sets, satisfying: i) $\mathbf{X}$ and the empty set belongs to open sets. ii) The union of any family of open sets lies in open sets. iii) The intersection of any finite family of open sets belongs to open sets.

**Hausdorff:** A topological space is said to be Hausdorff if for every pair of points $p, q \in \mathbf{X}$ there exists a disjoint neighborhood $\mathcal{U}$ and $\mathcal{V}$ of $p$ and $q$ respectively.i.e. $\mathcal{U} \cap \mathcal{V} = \varnothing$.

**Homeomorphism:** Given two topological spaces $\mathbf{X}$ and $\mathbf{Y}$, a mapping $f$ is called a homeomorphism, is it is bijective (one-to-one and onto) and $f : \mathbf{X} \to \mathbf{Y}$ and $f^{-1} : \mathbf{Y} \to \mathbf{X}$ are both continuous.

### 2.1.1 Coordinate Chart, Transition Map and Atlas

The coordinate chart is the neighborhood $\mathcal{U} \in \mathcal{M}$ and its associated homeomorphism $\phi$ from $\mathcal{U}$ to $\mathbb{R}^d$. The transition map $\phi \circ \psi^{-1}$ is a mapping from the open set $\psi(\mathcal{U} \cap \mathcal{V}) \in \mathbb{R}^d$ to the open set $\phi(\mathcal{U} \cap \mathcal{V}) \in \mathbb{R}^d$, where $(\mathcal{U}, \phi)$ and $(\mathcal{V}, \psi)$ are two different $d$-dimensional coordinate charts such that $(\mathcal{U} \cap \mathcal{V} \neq 0)$. If transition map for all coordinates chart is analytic i.e. has a convergent Taylor series expansion, the manifold is called analytic manifold and it is smooth and differentiable. An atlas is a set of coordinate such that $\phi_i(\mathcal{U}_i) \subseteq \mathbb{R}^d$ for all $i$; $\mathcal{U}_i$ covers $\mathcal{M}$ i.e. $\mathcal{M} = \bigcup_i \mathcal{U}_i$; transition maps $\phi_j \circ \phi_i^{-1}$ are smooth when $(\mathcal{U}_i \cap \mathcal{U}_j \neq 0)$. These ideas are graphically illustrated in Fig. 2.1.1.

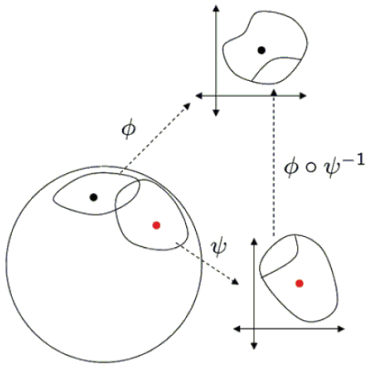


Figure 2.1: Two overlapping coordinate charts and transition map for two-dimensional manifold. The figure is taken from [46].

### 2.1.2 Functions on Manifold

Consider a real valued function $f : \mathcal{M} \to \mathbb{R}$ on the manifold. If for every $p \in \mathcal{M}$, there exists a chart $(\mathcal{U}, \phi)$ then the composite function $\hat{f} = f \circ \phi^{-1}$ which maps the open set $\phi(\mathcal{U}) \in \mathbb{R}^d$ to $\mathbb{R}$, is called coordinate representation of $f$. By definition, $f$ is smooth and analytic if the coordinate representation is smooth and analytic for all coordinate charts. The process of finding partial differentiation of $f$ at $\mathbf{p}$ on $\mathcal{M}$ involves pulling the function back to $\mathbb{R}^d$ by using the coordinate chart $(\mathcal{U}, \phi)$ and then evaluating the directional derivative. It can be expressed mathematically as:

$$\partial_i(f) = \frac{\partial(f \circ \phi^{-1})}{\partial u^i}|_{\phi(\mathbf{p})} \tag{2.1}$$

where $u^i$ is the $i$-th coordinate of the point $\phi(\mathbf{p})$ in $\mathbb{R}^d$.

### 2.1.3 Tangent Vectors on Manifold

In literature [46, 49, 50], the notion of tangent vectors at a point on a manifold is defined in several different ways. However, the most intuitive method to describe tangent vector is to use curves on manifold (page 34 [49]). A curve $\gamma$ in $\mathcal{M}$ is a smooth mapping from open interval in $\mathbb{R}$ to $\mathcal{M}$ i.e. $\gamma : \mathbb{R} \to \mathcal{M}$. A tangent vector $\Delta$ at a point $\mathbf{p}$ on a manifold $\mathcal{M}$ is a mapping from the set of smooth real-valued functions defined on a neighborhood of $\mathbf{p}$ (i.e $(\mathfrak{F}_{\mathbf{p}}(\mathcal{M}))$) to $\mathbb{R}$ such that there exists a curve $\gamma$ on $\mathcal{M}$ with $\gamma(0) = \mathbf{p}$, satisfying:

$$\Delta(f) = (\dot{\gamma})(0)(f) := \frac{d(f(\gamma(t)))}{dt}|_{t=0} \tag{2.2}$$

for all $f \in \mathfrak{F}_{\mathbf{p}}\mathcal{M})$. The curve $\gamma$ is then said to realize the tangent vector $\Delta$ and the point $\mathbf{p}$ is called foot of the tangent vector. However, there are infinitely many curves $\gamma$ that realize $\Delta$ (i.e. $\Delta = \dot{\gamma}(0)$). Two curves $\gamma_1$ and $\gamma_2$ through a point $\mathbf{p}$ at $t = 0$ satisfy $\dot{\gamma}_1(0) = \dot{\gamma}_2(0)$, if given a chart $(\mathcal{U}, \phi)$ with $\mathbf{p} \in \mathcal{U}$, it holds that:

$$\frac{d(\phi(\gamma_1(t)))}{dt}|_{t=0} = \frac{d(\phi(\gamma_2(t)))}{dt}|_{t=0} \tag{2.3}$$

A tangent vector on $\mathcal{M}$ at $\mathbf{p}$ can also be viewed as a real-valued operator [46] on continuous function satisfying the properties of linearity and Leibniz product rule of derivative:

$$\Delta(af + bh) = a\Delta(f) + b\Delta(h) \tag{2.4}$$
$$\Delta(fh) = f\Delta(h) + h\Delta(f) \tag{2.5}$$

for all continuous functions $f, h$ and $a, b \in \mathbb{R}$.
The set of all tangent vectors to $\mathcal{M}$ at $\mathbf{p}$ is called tangent space, denoted by $\mathcal{T}_{\mathbf{p}}$. It is a vector space satisfying the rules of addition and scalar multiplication:

$$(\Delta + \Gamma)(f) = \Delta(f) + \Gamma(f) \tag{2.6}$$
$$(a\Delta)(f) = a\Delta(f) \tag{2.7}$$

Intuitively, the tangent space can be thought of as the set of allowed velocities for a point constrained to move on manifold. It gives a linear approximation for the manifold at a point.
For $d$-dimensional manifold, the dimension of tangent space is equal to $d$. if $(\mathcal{U}, \phi)$ is a chart containing $\mathbf{p}$ and $u^i$ is the $i^{th}$ coordinate of the point $\phi(\mathbf{p})$, then $\frac{\partial(f \circ \phi^{-1})}{\partial u^i}|_{\phi(\mathbf{p})}$, $i = 1, \ldots, d$ form the basis of $\mathcal{T}_{\mathbf{p}}$.

To define a notion of length of tangent vectors, the tangent space at every point $\mathbf{p}$ on a manifold is associated with an inner product $<,.,>_p$. Given a set of basis vectors for the tangent space $\mathcal{T}_{\mathbf{p}}$, $<,.,>_p$ can always be represented as a symmetric positive definite matrix. The inner product induces a norm $||\Delta||_{\mathbf{p}} \triangleq \sqrt{<\Delta, \Delta>_{\mathbf{p}}}$ on $\mathcal{T}_{\mathbf{p}}$. A manifold whose tangent spaces are endowed with a smoothly varying inner product is called Riemannian manifold. The smoothly varying inner product is called Riemannian metric. Two different inner product metrics on the same manifold would lead to two different Riemannian manifolds. However, in practise there exists a standard metric and the Riemannian manifold is denoted ny underlying analytic manifold. Given two tangent vectors $\Delta, \Gamma \in \mathcal{T}_{\mathbf{p}}$, their inner product is written as $<\Delta, \Gamma>$. This enables to define angles between the two tangent vectors.

### 2.1.4 Distances on Manifold

The distance between two points $\mathbf{p}$ and $\mathbf{q}$ on the manifold is defined by integrating over all piecewise smooth curve segments from $\mathbf{p}$ to $\mathbf{q}$.
For a closed interval $[a, b] \subseteq \mathbb{R}$, a map $\gamma : [a, b] \to \mathcal{M}$ is a piecewise smooth curve from $\mathbf{p} = \gamma(a)$ to $\mathbf{q} = \gamma(b)$, if there is a sequence of numbers $a = n_0 < n_1 < \ldots < n_{i-1} < n_i = b; n_i \in \mathbb{R}$ so that each map $\gamma : [n_j \ n_{j+1}]$ is a smooth curve for $j = 0, \ldots, i-1$
The length of curve $\gamma$ between $p$ and $q$ is given as ([48] page 374):

$$L(\gamma) = \sum_{i=0}^{k-1} \int_{t \in_{t_i}^{t_{i+1}}} \sqrt{<\gamma'(t), \gamma'(t)>} dt \tag{2.8}$$

where $\gamma'(t)$ is the tangent at $\gamma(t)$.

A curve on manifold is a geodesic if it is the shortest path connecting two points. The length of geodesic is called Riemannian distance between two points and the velocity is constant along geodesic.

### 2.1.5 Exponential and Logarithmic Mapping Functions

These functions map $\mathbf{p} \in \mathcal{M}$, along geodesics, from tangent space $\mathcal{T}_{\mathbf{p}}$ to the manifold and vice versa. For each $\mathbf{\Delta} \in \mathcal{T}_{\mathbf{p}}$, there exists a locally unique geodesic $\gamma(t)$ starting at $\gamma(0) = \mathbf{p}$ with initial velocity $\dot{\gamma}(0) = \mathbf{\Delta}$ and traveling with constant speed.
The exponential mapping function $\exp_{\mathbf{p}} : \mathcal{T}_{\mathbf{p}} \to \mathcal{M}$ maps the tangent vector $\mathbf{\Delta}$ along the geodesic $\gamma(t)$ to a point $\mathbf{q}$ on the manifold that is reached in unit time i.e. $\mathbf{q} = \exp_{\mathbf{p}}(\mathbf{\Delta}) = \gamma(1)$.
The exponential map defines a diffeomorphism (smooth bijection) of a neighborhood $\tilde{\mathcal{U}}$ of the origin at $T_{\mathbf{p}}$ to a neighborhood $\mathcal{U}$ of $\mathbf{p}$ in the manifold. The inverse of the

exponential map i.e. a mapping from $\mathcal{U}$ to $\tilde{\mathcal{U}}$ exists locally in the neighborhood of $\mathbf{p}$ and is called logarithmic map $\log_{\mathbf{p}} : \mathcal{M} \to T_{\mathbf{p}}$.
It is worth mentioning, that these operators depend on the base point on the manifold and change as the point moves. This fact is represented explicitly as subscript in the notation. Moreover, the exponential and logarithmic maps are different for each manifold and for each metric. Thus they have to be determined and implemented on a case by case basis. The specific formula for certain manifolds are presented in Section 2.2-2.5. Fig. 2.2 shows a two dimensional manifold in $\mathbb{R}^3$.

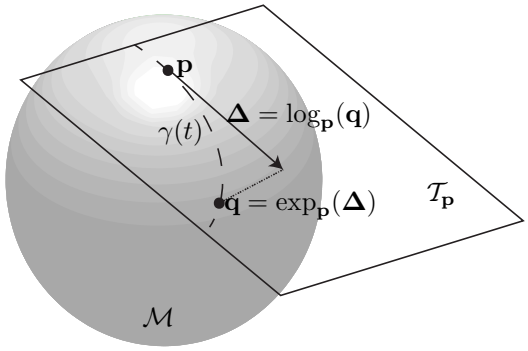


Figure 2.2: An example of a two-dimensional manifold $\mathcal{M}$ embedded in $\mathbb{R}^3$. $\mathbf{p}$ and $\mathbf{q}$ are manifold points, $T_{\mathbf{p}}\mathcal{M}$ is the tangent plane for $\mathbf{p}$, where $\boldsymbol{\Delta}$ is the tangent vector originated from $\mathbf{p}$ and end in $\mathbf{q}$. The geodesic $\gamma(t)$ is the shortest curve between $\mathbf{p}$ and $\mathbf{q}$ on the manifold.

## 2.2 Lie Groups

A Group $G$ is a set of elements with an operator '.', such that the following properties are satisfied: i)closure; for any $g_1, g_2 \in G$ implies $g_1.g_2 \in G$. ii) associative; for any $g_1, g_2, g_3 \in G$ implies $g_1.(g_2.g_3) = (g_1.g_2).g_)$. iii) identity; for any $g_1 \in G$ there exists $e \in G$, such that $g_1.e = g_1$. iv) inverse; for any $g_1 \in G$ there exists $g_1^{-1} \in G$, such that $g_1.g_1^{-1} = e$.

A Lie Group is a group $\mathcal{G}$ with the structure of smooth manifold such that multiplication map $m : \mathcal{G} \times \mathcal{G} \to \mathcal{G}$ ( given by $m(g_1, g_2) = g_1 g_2$ for any $g_1, g_2 \in \mathcal{G}$) and inversion map $i : \mathcal{G} \to \mathcal{G}$ ( given by $i(g_1) = g_1^{-1}$ $g_1, g_2 \in \mathcal{G}$) are analytic. The group operation gives Lie groups with additional group structure. The tangent space at the identity element of the group forms a Lie algebra, denoted by $\mathfrak{g}$, which is a vector space that is closed under the Lie bracket operation. A Lie bracket is a bilinear operation that satisfies the identity of bilinearity, antisymmetry and Jacobi symmetry (see p.94 of [50] for more details).
The most commonly used Lie groups are matrix Lie groups, also called as general linear group $\mathbf{GL}(n)$. These are sets of $n \times n$ non-singular matrices and the group operation is matrix multiplication. Some commonly used examples of matrix Lie groups include: i) The orthogonal group $\mathbf{O}(n)$ is the set of real $n \times n$ orthogonal

matrices. ii) special orthogonal group (or group of rotations) $\mathbf{SO}(n)$ is the subgroup of $\mathbf{O}(n)$ consisting of those matrices of $\mathbf{O}(n)$ having determinant $+1$. (iii) The special Euclidean group $\mathbf{SE}(n)$ is the group of affine maps of $\mathbb{R}^n$ in terms of $(n+1)\times(n+1)$ matrices of the form $\begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}$, where $\mathbf{R} \in \mathbf{SO}(n)$ and $\mathbf{t} \in \mathbb{R}^n$. The formula for exponential map, log map and distance function are (see [52] Chap. 11):

$$\exp_{\mathbf{p}}(\mathbf{\Delta}) = \mathbf{p}\exp(\mathbf{p}^{-1}\mathbf{\Delta}) = \mathbf{q} \tag{2.9}$$

$$\log_{\mathbf{p}}(\mathbf{q}) = \mathbf{p}\log(\mathbf{p}^{-1}\mathbf{q}) = \mathbf{\Delta} \tag{2.10}$$

$$d(\mathbf{p},\mathbf{q}) = ||\log(\mathbf{p}^{-1}\mathbf{q})||_F \tag{2.11}$$

where $\mathbf{p},\mathbf{q} \in \mathcal{M}$, $\mathbf{\Delta} \in T_{\mathbf{p}}\mathcal{M}$ and $||.||_F$ denotes the Frobenius norm of matrix.

## 2.3 Geodesic on Orthogonal Group

This section reviews the derivation of geodesic on Lie groups of orthogonal matrices. These derivations are required later for Grassmann manifolds. Moreover these equations will be required later for Grassmann manifold. Readers may refer to [37] for further information.

Let $\mathbf{Q}$ be an $n \times n$ orthogonal matrix represented as a point on the orthogonal group $\mathbf{O}(n)$ then the condition of orthogonality is given as:

$$\mathbf{Q}^T\mathbf{Q} = \mathbf{I}_{n\times n} \tag{2.12}$$

To find the geodesic equation for a curve $\gamma(t) = \mathbf{Q}_t$ between the two points $\mathbf{Q}_0, \mathbf{Q}_1 \in \mathbf{O}(n)$ such that $\gamma(0) = \mathbf{Q}_0$ and $\gamma(t) = \mathbf{Q}_1$ where $0 \le t \le t_1, t \in \mathbb{R}$, orthogonality constraint $\mathbf{Q}_t^T\mathbf{Q}_t = \mathbf{I}$ is differentiated with respect to $t$ twice:

$$\mathbf{Q}_t^T\dot{\mathbf{Q}}_t + \dot{\mathbf{Q}}_t^T\mathbf{Q}_t = 0 \tag{2.13}$$

$$\mathbf{Q}_t^T\ddot{\mathbf{Q}}_t + 2\dot{\mathbf{Q}}_t^T\dot{\mathbf{Q}}_t + \ddot{\mathbf{Q}}_t^T\mathbf{Q}_t = 0 \tag{2.14}$$

where $\dot{\mathbf{Q}}_t$ and $\ddot{\mathbf{Q}}_t$ are the instantaneous velocity and acceleration at $\mathbf{Q}_t$.

Tangent space $\mathcal{T}_{\mathbf{Q}}$ is a set of all first derivatives at $\mathbf{Q}$. It can be expresses by using $\mathbf{\Delta} = \dot{\mathbf{Q}}$ in (2.13) as:

$$\mathcal{T}_{\mathbf{Q}} = \{\mathbf{\Delta} : \mathbf{Q}^T\mathbf{\Delta} + \mathbf{\Delta}^T\mathbf{Q} = 0\} \tag{2.15}$$

It is the space of skew-symmetric matrices explained as: Let $\mathbf{K} = \mathbf{Q}^T\mathbf{\Delta}$, then $\mathbf{\Delta} = \mathbf{Q}\mathbf{K}$. Using $\mathbf{\Delta}$ in (2.15), $\mathbf{Q}^T(\mathbf{Q}_t\mathbf{K}) + (\mathbf{Q}^T\mathbf{K}^T)\mathbf{Q} = \mathbf{K} + \mathbf{K}^T = 0$, so matrix $\mathbf{K}$ must be skew-symmetric.

Normal space $\mathcal{N}_{\mathbf{Q}}$ at at $\mathbf{Q}$ is the orthogonal complement of the tangent space at $\mathbf{Q}$. The orthogonal complement of the skew symmetric tangent space is the space of symmetric matrices, explained for the $2\times 2$ matrix as follows: Let $\mathbf{S} = \begin{bmatrix} a & b \\ b & c \end{bmatrix}$ be the symmetric matrix. For $\mathbf{K} = \begin{bmatrix} w & x \\ y & z \end{bmatrix}$ to be the orthogonal complement of $\mathbf{S}$,

the condition of orthogonality $tr(\mathbf{S}^T\mathbf{K}) = 0$ is satisfied if $aw + b(x + y) + cz = 0$ for any $a, b, c$. Thus $w = z = (x + y) = 0$ and $\mathbf{K} = \begin{bmatrix} 0 & -y \\ y & 0 \end{bmatrix}$ is skew symmetric.
Thus we can write the tangent space and its normal space as:

$$\mathcal{T}_{\mathbf{Q}} = \{\mathbf{\Delta} : \mathbf{\Delta} = \mathbf{QK},\ \mathbf{K}^T + \mathbf{K} = 0\} \tag{2.16}$$
$$\mathcal{N}_{\mathbf{Q}} = \{\mathbf{N} : \mathbf{N} = \mathbf{QS},\ \mathbf{S}^T = \mathbf{S}\} \tag{2.17}$$

The necessary and sufficient condition to be a geodesic is that the acceleration vector $\ddot{\mathbf{Q}}_\mathbf{t}$ at $\mathbf{Q}_t$ is in normal space yielding:

$$\ddot{\mathbf{Q}}_t + \mathbf{Q}_t\mathbf{S} = 0 \tag{2.18}$$

Using $\ddot{\mathbf{Q}}_t = -\mathbf{Q}_t\mathbf{S}$ in (2.14):

$$\mathbf{Q}_t^T(-\mathbf{Q}_t\mathbf{S}) + 2\dot{\mathbf{Q}}_t^T\dot{\mathbf{Q}}_t + (-\mathbf{Q}_t\mathbf{S})^T\mathbf{Q}_t = 0 \tag{2.19}$$
$$-\mathbf{S} + 2\dot{\mathbf{Q}}_t^T\dot{\mathbf{Q}}_t + \mathbf{S}^T = 0 \tag{2.20}$$
$$\mathbf{S} = \dot{\mathbf{Q}}_t^T\dot{\mathbf{Q}}_t \tag{2.21}$$

The facts ($\mathbf{Q}_t^T\mathbf{Q}_t = \mathbf{I}$) and $\mathbf{S} = \mathbf{S}^T =$ are used for simplification of (2.19) and (2.20) respectively. Putting (2.21) in (2.18), we get:

$$\ddot{\mathbf{Q}}_t + \mathbf{Q}_t\dot{\mathbf{Q}}_t^T\dot{\mathbf{Q}}_t = 0 \tag{2.22}$$

From (2.16), $\mathbf{\Delta} = \mathbf{QK}$, the solution for geodesic equation can be obtained as follows:

$$\dot{\mathbf{Q}}_\mathbf{t} = \mathbf{Q}_t\mathbf{K} \tag{2.23}$$
$$\dot{\mathbf{Q}}_\mathbf{t} - \mathbf{Q}_t\mathbf{K} = 0 \tag{2.24}$$

Multiplying both sides of (2.24) by integrating factor $e^{\int -\mathbf{K}.dt} = e^{-\mathbf{K}t}$ and using the differentiation chain rule, we get:

$$e^{-\mathbf{K}t}\dot{\mathbf{Q}}_\mathbf{t} - e^{-\mathbf{K}t}\mathbf{Q}_t\mathbf{K} = 0 \tag{2.25}$$
$$(e^{-\mathbf{K}t}\mathbf{Q}_t)' = 0 \tag{2.26}$$

Integrating both sides of (2.26) with respect to $t$, we get:

$$(e^{-\mathbf{K}t}\mathbf{Q}_t) = c \tag{2.27}$$
$$\mathbf{Q}_t = ce^{-\mathbf{K}t} \tag{2.28}$$

for $t = 0$, $\mathbf{Q}_0 = c$, thus (2.28) becomes:

$$\mathbf{Q}_t = \mathbf{Q}_0e^{-\mathbf{K}t} \tag{2.29}$$

## 2.4 Grassmann Manifold

A Grassmann manifold $\mathcal{G}_{n,k}$ is a set of all $k$-dimensional subspaces in $\mathbb{R}^n$. A point on $\mathcal{G}_{n,k}$ can be represented as quotient space within the orthogonal group $\mathbf{O}(n)$ by using

the equivalence classes. Thus given a point $\mathbf{Q} \in \mathbf{O}(n)$, its equivalence class $[\mathbf{Q}]$ for the Grassmann manifold $\mathcal{G}_{n,k}$ is given as:

$$[\mathbf{Q}] = \left\{ \mathbf{Q} \begin{bmatrix} \mathbf{Q}_k & 0 \\ 0 & \mathbf{Q}_{n-k} \end{bmatrix} : \begin{pmatrix} \mathbf{Q}_k \in \mathbf{O}(k) \\ \mathbf{Q}_{n-k} \in \mathbf{O}(n-k) \end{pmatrix} \right\} \tag{2.30}$$

Two matrices are equivalent if their columns span the same $k$-dimensional subspace in $\mathbb{R}^n$. The tangent space at point $\mathbf{Q}$ can be expressed by horizontal and vertical spaces which are orthogonal complement of each other. The horizontal space at $\mathbf{Q}$ is the set of tangents of the form:

$$\boldsymbol{\Delta}_{\mathbf{Q}} = \mathbf{QK} = \mathbf{Q} \begin{bmatrix} 0 & -\mathbf{B}^T \\ \mathbf{B} & 0 \end{bmatrix} \tag{2.31}$$

where $\mathbf{K}$ is the is the $n \times n$ skew-symmetric matrix and $\mathbf{B}$ is any arbitrary $(n-k) \times n$ matrix. The vertical space is defined as set of vectors tangent to the entire equivalence class $[\mathbf{Q}]$.

A point on the Grassmann manifold can also be represented by $n \times k$ orthonormal basis matrix $\mathbf{U}$. The left coset or equivalence class or orbit of $\mathbf{O}_k$ with respect to matrix $\mathbf{U}$ is:

$$[\mathbf{U}] = \{\mathbf{UQ}_k : \mathbf{Q}_k \in \mathbf{O}(k)\} \tag{2.32}$$

Another strategy to represent a point in the Grassmann manifold is by $n \times n$ projection matrix $\mathbf{U}(\mathbf{U^T U})^{-1}\mathbf{U^T}$. In this work, $n \times k$ matrix representation is used due to computational efficiency.

To form a full basis for $\mathbb{R}^n$, an $n \times n$ orthonormal matrix $\mathbf{Q}$ can be defined as $\mathbf{Q} \triangleq [\mathbf{U}|\mathbf{U}_\perp]$ , where $\mathbf{U}_\perp$ is an $n \times (n-k)$ orthonormal complement of $\mathbf{U}$ such that $\mathbf{U}^T\mathbf{U}_\perp = 0$. Tangent space at $\mathbf{U}$ consists of tangent vector $\boldsymbol{\Delta}_{\mathbf{U}}$ (of the form (2.33) ) such that $\mathbf{U}^T\boldsymbol{\Delta}_{\mathbf{U}} = 0$.

$$\boldsymbol{\Delta}_{\mathbf{U}} = [\mathbf{U} \;\; \mathbf{U}_\perp] \begin{bmatrix} 0 & -\mathbf{B}^T \\ \mathbf{B} & 0 \end{bmatrix} \mathbf{I}_{n \times k} = \mathbf{U}_\perp \mathbf{B} = \mathbf{H} \tag{2.33}$$

where $\boldsymbol{\Delta}_{\mathbf{U}} = \mathbf{H}$ is the tangent vector at $\mathbf{U}$.

The geodesic for the Grassmann manifold, by using (2.30), is given as:

$$\mathbf{U}_t = \mathbf{Q}_0 e^{-\mathbf{K}t} \mathbf{I}_{n \times k} \tag{2.34}$$

The formula for computing geodesic is: (Theorem (2.3) of [37]):

$$\mathbf{U}_t = (\mathbf{UV} \quad \mathbf{R}) \begin{pmatrix} \cos \boldsymbol{\Sigma} t \\ \sin \boldsymbol{\Sigma} t \end{pmatrix} \mathbf{V}^T \tag{2.35}$$

where $\mathbf{R\Sigma V}$ is the compact singular value decomposition of $\mathbf{H} = \mathbf{U}_\perp \mathbf{B}$ and the sin and cos act element-by-element along the diagonal $\Sigma$.

**Proof:** We decompose $\mathbf{Q}, \boldsymbol{\Delta}$ and skew-symmetric matrix $\mathbf{K}$ into block matrices as follows:

$$\mathbf{Q}_{n \times n} = [\mathbf{U}_{n \times k} \;\; \mathbf{U}_{\perp (n \times n-k)}] \tag{2.36}$$

$$\boldsymbol{\Delta}_{n \times n} = [\mathbf{H}_{n \times k} \;\; \mathbf{H}_{\perp (n \times n-k)}] \tag{2.37}$$

$$\mathbf{K}_{n \times n} = \begin{bmatrix} 0 & -\mathbf{B}^T_{k \times n-k} \\ \mathbf{B}_{n-k \times k} & 0 \end{bmatrix} \tag{2.38}$$

Let us decompose $\mathbf{B}$ in (2.37) by using an SVD decomposition as follows:

$$\begin{aligned}\mathbf{B} &= \begin{bmatrix} \mathbf{U}_{1(n-k\times k)} & \mathbf{U}_{2(n-k\times n-2k)} \end{bmatrix} \begin{bmatrix} \mathbf{\Sigma}_{k\times k} \\ \mathbf{0}_{n-2k\times k} \end{bmatrix} \mathbf{V}^T_{k\times k} && (2.39)\\ &= \mathbf{U}_1\mathbf{\Sigma}\mathbf{V}^T && (2.40)\end{aligned}$$

Substituting (2.39) in (2.38), we get:

$$\begin{aligned}\mathbf{K} &= \begin{bmatrix} \mathbf{0} & \mathbf{V}(\mathbf{\Sigma}\mathbf{U}_1^T + \mathbf{0}.\mathbf{U}_2^T) \\ (\mathbf{U}_1\mathbf{\Sigma} + \mathbf{U}_2.\mathbf{0})\mathbf{V}^T & \mathbf{0} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{V} & 0 & 0 \\ 0 & \mathbf{U}_1 & \mathbf{U}_2 \end{bmatrix} \begin{bmatrix} \mathbf{0} & -\mathbf{\Sigma}\mathbf{U}_1^T \\ \mathbf{\Sigma}\mathbf{V}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{V} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{U}_1 & \mathbf{U}_2 \end{bmatrix} \begin{bmatrix} \mathbf{0} & -\mathbf{\Sigma} & \mathbf{0} \\ \mathbf{\Sigma} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{V}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{U}_1^T \\ \mathbf{0} & \mathbf{U}_2^T \end{bmatrix} && (2.41)\end{aligned}$$

Using the facts (Chapter 1 [48]) $e^{\mathbf{AXA}^{-1}} = \mathbf{A}e^{\mathbf{X}}\mathbf{A}^{-1}$ and $\exp\begin{bmatrix} 0 & -\theta \\ \theta & 0 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$,(2.41) can be written as:

$$e^{\mathbf{K}t} = \begin{bmatrix} \mathbf{V} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{U}_1 & \mathbf{U}_2 \end{bmatrix} \begin{bmatrix} c & -s & \mathbf{0} \\ s & c & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{V}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{U}_1^T \\ \mathbf{0} & \mathbf{U}_2^T \end{bmatrix} \quad (2.42)$$

where $c = \cos(\mathbf{\Sigma}\mathbf{t})$ and $s = \sin(\mathbf{\Sigma}\mathbf{t})$. Multiplying both sides of (2.42) by $\mathbf{Q}_0$, we get:

$$\begin{aligned}\mathbf{Q}_0 e^{\mathbf{K}t} &= [\mathbf{U}_0 \;\; \mathbf{U}_{\perp 0}] \begin{bmatrix} \mathbf{V}c\mathbf{V}^T & -\mathbf{V}s\mathbf{U}^T \\ \mathbf{U}_1 s\mathbf{V}^T & \mathbf{U}_1 c\mathbf{U}_1^T + \mathbf{U}_2\mathbf{U}_2^T \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{U}_0\mathbf{V}c\mathbf{V}^T + \mathbf{U}_{\perp 0}\mathbf{U}_1 s\mathbf{V}^T \\ -\mathbf{U}_0\mathbf{V}s\mathbf{U}_1^T + \mathbf{U}_{\perp 0}\mathbf{U}_1 c\mathbf{U}_1^T + \mathbf{U}_{\perp 0}\mathbf{U}_2\mathbf{U}_2^T \end{bmatrix}^T && (2.43)\end{aligned}$$

Constraining to work with subspace spanned by first $k$-columns given by $\mathbf{U}_0$, equation for geodesic can be obtained as:

$$\mathbf{U}_t = \begin{bmatrix} \mathbf{U}_0\mathbf{V}c\mathbf{V}^T + \mathbf{U}_{\perp 0}\mathbf{U}_1 s\mathbf{V}^T \end{bmatrix} \quad (2.44)$$

From (2.31), we get: $\mathbf{\Delta}_0 = \mathbf{Q}_0\mathbf{K}$. By using (2.36) and (2.37):

$$\begin{aligned}[\mathbf{H}_0 \;\; \mathbf{H}_{\perp 0}] &= [\mathbf{U}_0 \;\; \mathbf{U}_{\perp 0}] \begin{bmatrix} 0 & -\mathbf{B}^T_{k\times n-k} \\ \mathbf{B}_{n-k\times k} & 0 \end{bmatrix} \\ &= [\mathbf{U}_{\perp 0}\mathbf{B} \mid \mathbf{U}_0\mathbf{B}^T] && (2.45)\end{aligned}$$

Again by Constraining to work with subspace spanned by first $k$-columns and using (2.37) and (2.40), we get $\mathbf{H}_0 = \mathbf{U}_{\perp 0}\mathbf{B} = \mathbf{U}_{\perp 0}[\mathbf{U}_1\mathbf{\Sigma}\mathbf{V}^T]$. Substituting $\mathbf{U}_{\perp 0}\mathbf{U}_1 = \mathbf{H}_0\mathbf{\Sigma}^{-1}\mathbf{V}$ in (2.44), we get:

$$\mathbf{U}_t = \left[ \mathbf{U}_0\mathbf{V}c\mathbf{V}^T + \mathbf{H}_0\mathbf{\Sigma}^{-1}\mathbf{V}s\mathbf{V}^T \right] \tag{2.46}$$

Finally by using the initial condition, $\mathbf{U}_0 = \mathbf{U}, \mathbf{H}_0 = \mathbf{H}$ and compact SVD of $\mathbf{H}$ as $\mathbf{H} = \mathbf{S}\mathbf{\Sigma}\mathbf{V}^T$ in (2.46), we obtain:

$$\mathbf{U}_t = \left[ \mathbf{U}\mathbf{V}c\mathbf{V}^T + \mathbf{S}s\mathbf{V}^T \right] \tag{2.47}$$

$$= [\mathbf{U}\mathbf{V} \;\; \mathbf{S}] \begin{bmatrix} \cos(\mathbf{\Sigma}\mathbf{t}) \\ \sin(\mathbf{\Sigma}\mathbf{t}) \end{bmatrix} \mathbf{V}^T \tag{2.48}$$

□

### 2.4.1 Exponential Map

The exponential map of the tangent vector $\mathbf{\Delta}$ at $\mathbf{p}$ to yield the end point $\mathbf{q}$ along the geodesic can be obtained by putting $t = 1$ in (2.35) as:

$$\exp_{\mathbf{p}}(\Delta) = \mathbf{U}(1) = \mathbf{W} = \mathbf{U}\mathbf{V}\cos(\mathbf{\Sigma})\mathbf{V}^T + \mathbf{R}\sin(\mathbf{\Sigma})\mathbf{V}^T \tag{2.49}$$

where $\mathbf{p}, \mathbf{q} \in \mathcal{G}_{n,k}$, $\mathbf{U}, \mathbf{W}$ are the $n \times k$ orthonormal bases matrix of $\mathbf{p}$ and $\mathbf{q}$ respectively.

### 2.4.2 Logarithmic Map

The logarithmic map can be defined in many equivalent ways [46, 53–55]. In this work, we have used the formula proposed by [46]:

$$\mathbf{\Delta} = \log_{\mathbf{p}}(\mathbf{q}) = \mathbf{S}\sin^{-1}(\mathbf{\Sigma})\mathbf{V}^T \tag{2.50}$$

where $\mathbf{p}, \mathbf{q} \in \mathcal{G}_{n,k}$, $\mathbf{U}, \mathbf{W}$ are the $n \times k$ orthonormal bases matrix of $\mathbf{p}$ and $\mathbf{q}$ respectively, $\mathbf{R}\mathbf{\Sigma}\mathbf{D}^T = \mathbf{W} - \mathbf{U}\mathbf{U}^T\mathbf{W}$ and $\mathbf{V}\mathbf{C}\mathbf{D}^T = \mathbf{U}^T\mathbf{W}$ is the generalized SVD with $\mathbf{C}^T\mathbf{C} + \mathbf{\Sigma}^T\mathbf{\Sigma} = I$ and the $\sin^{-1}$ acts element-by-element along the diagonal of $\mathbf{\Sigma}$. The two mapping functions satisfy $\mathbf{U}^T\log_{\mathbf{U}}(\mathbf{W}) = 0$ and $\exp_{\mathbf{U}}(\log_{\mathbf{U}}(\mathbf{W})) = \mathbf{W}$.

### 2.4.3 Distance

The Riemannian distance or arc length between $\mathbf{p}$ and $\mathbf{q}$ can be calculated by the set of principal angles [37] as follows:

$$d(\mathbf{p}, \mathbf{q}) = \|\theta\|_2 \tag{2.51}$$

where $\theta$ (the principal angle) can be obtained by first calculating the compact singular value decomposition $\mathbf{R}\mathbf{\Sigma}\mathbf{D}^T$ of $\mathbf{U}^T\mathbf{W}$ and then finding $\cos^{-1}(\sigma_i^2)$ where $\sigma_1^2, \ldots, \sigma_k^2$ are the diagonal values of $\mathbf{\Sigma}$. $\mathbf{U}, \mathbf{W}$ are the $n \times k$ orthonormal bases matrix of $\mathbf{p}$ and $\mathbf{q}$ respectively.

## 2.5 Symmetric Manifold

The set of $n \times n$ symmetric positive definite matrices forms a manifold known as symmetric manifold $symm_n^+$. Positive definite matrices symmetric are used in many applications of image and video processing. e.g To encode the Brownian motion (diffusion) of water in Diffusion Tensor Imaging (DTI); to encode the joint variability at different places in shape analysis; to guide the segmentation, grouping and motion analysis in image analysis; to encode image features in video tracking and many more. The space of Symmetric Positive Definite matrices is not a vector space (not closed w.r.t multiplication by negative scalar). Instead, it lies on Riemannian manifold. The tangent space at a point $\mathbf{p}$ on the manifold consists of symmetric matrices, not necessarily symmetric positive definite matrices.

Two Riemannian metrics, namely affine-invariant metric and Log-Euclidean metric, proposed by [56] and [57], are used for computing statistics on this manifold. Numerical results of both Riemannian metrics are similar. However, the Log-Euclidean metric is computationally efficient and calculation of mean points on $\mathcal{M}$ is easy with a closed form.

The affine invariant metric [56] defines action of the linear group $\mathbf{GL}_n$ (affine group) on the tensor space and provides an invariant Riemannian metric with respect to it. The formula for exponential, logarithmic map and distance for this metric are:

$$\begin{array}{l} \exp_{\mathbf{p}}(\boldsymbol{\Delta}) = \mathbf{p}^{1/2} \exp(\mathbf{p}^{-1/2}\boldsymbol{\Delta}\mathbf{p}^{-1/2})\mathbf{p}^{1/2}; \\ \log_{\mathbf{p}}(\mathbf{q}) = \mathbf{p}^{1/2} \log(\mathbf{p}^{-1/2}\mathbf{q}\mathbf{p}^{-1/2})\mathbf{p}^{1/2} \\ d(\mathbf{p},\mathbf{q}) = ||\log(\mathbf{p}^{1/2}\mathbf{q}\mathbf{p}^{-1/2})||_{Id}^2 \end{array} \tag{2.52}$$

where $\mathbf{p}, \mathbf{q}$ are the points on manifold and $\boldsymbol{\Delta}$ is the velocity vector corresponding to the geodesic from $\mathbf{p}$ to $\mathbf{q}$. Refer to [56] for further details.

The Log-Euclidean metric [57] uses the fact that matrix exponential is a diffeomorphism from the space of symmetric matrices to the tensor space. This provides tensor space with commutative Lie group structure with the matrix multiplication generalized by logarithmic multiplication. The exponential and logarithmic map associated with this metric can be expressed as matrix exponentials and logarithms in the following way:

$$\exp_{\mathbf{p}}(\boldsymbol{\Delta}) = \exp(\log \mathbf{p} + \boldsymbol{\Delta}) \tag{2.53}$$

$$\log_{\mathbf{p}} \mathbf{q} = \log \mathbf{q} - \log \mathbf{p} = \boldsymbol{\Delta} \tag{2.54}$$

Given two points $\mathbf{p}$, $\mathbf{q}$ on $\mathcal{M}$, the geodesic under Log-Euclidean metric is given by:

$$d(\mathbf{p},\mathbf{q}) = \| \log_{\mathbf{p}} \mathbf{q}\|_2 = \| \log \mathbf{q} - \log \mathbf{p}\|_2 \tag{2.55}$$

Readers are referred to [56, 57] for further details.

CHAPTER 3

# Related Methods of Visual Object Tracking

This chapter explains the concepts encompassed by the mean shift, local point features and particle filter framework used in this thesis. Section 2.1 describes Bayesian filtering, its general solution and possible ways to approximate this solution with emphasis on particle filter algorithm. Section 2.2 describes the basic theory of mean shift followed by an overview of visual tracking with isotropic and anisotropic kernel. Finally, section 2.3 describes the local point features tracking.

## 3.1 Sequential Bayesian Estimation

The aim of sequential Bayesian estimation is to find the posterior pdf $p(\mathbf{s}_t|\mathbf{z}_{1:t})$ of an underlying state vector $\mathbf{s}_t$ at the current time instance, using all available observations $\mathbf{z}_{1:t} = (\mathbf{z}_1, \ldots, \mathbf{z}_t)$ [58]. A common choice, for state $\mathbf{s}_t$ estimation, is the minimum mean-square error (MMSE) estimate:

$$\hat{\mathbf{s}}_t^{MMSE} = \mathbb{E}(\mathbf{s}_t|\mathbf{z}_{1:t}) = \int \mathbf{s}_t p(\mathbf{s}_t|\mathbf{z}_{1:t}) d\mathbf{s}_t \tag{3.1}$$

Another choice is the maximum-a-posteriori (MAP) estimate, measured as the state $\mathbf{s}_t$ that maximizes posterior $p(\mathbf{s}_t|\mathbf{z}_t)$, given by:

$$\hat{\mathbf{s}}_t^{MAP} = \arg\max_{\mathbf{s}_t} p(\mathbf{s}_t|\mathbf{z}_{1:t}) \tag{3.2}$$

Similarly, a measure of accuracy of state estimate (e.g. covariance) can also be obtained from $p(\mathbf{s}_t|\mathbf{z}_{1:t})$. In the next sections we first present the exact calculation of the posterior density and then its approximation by commonly used methods e.g. Kalman and particle filters.

### 3.1.1 Conceptual Solution

The exact solution of the posterior density estimation can be obtained by applying the Bayes theorem, the law of total probability and exploiting the following three naive independence assumptions:

- The probability of observation $\mathbf{z}_t$ conditioned on the current state $\mathbf{s}_t$ and previous history of observation $\mathbf{z}_{1:t-1}$ is independent of previous observations. i.e.

$$p(\mathbf{z}_t|\mathbf{s}_t, \mathbf{z}_{1:t-1}) = p(\mathbf{z}_t|\mathbf{s}_t) \tag{3.3}$$

- The probability of the current state $\mathbf{s}_t$ conditioned on the past state $\mathbf{s}_{t-1}$ and previous history of observation $\mathbf{z}_{1:t-1}$ is independent of the previous observations. i.e.

$$p(\mathbf{s}_t|\mathbf{s}_{t-1}, \mathbf{z}_{1:t-1}) = p(\mathbf{s}_t|\mathbf{s}_{t-1}) \tag{3.4}$$

- The probability of the current state depends only on the past state and is independent of the previous history of states or the 1st order Markov assumption. i.e.

$$p(\mathbf{s}_t|\mathbf{s}_{1:t-1}) = p(\mathbf{s}_t|\mathbf{s}_{t-1}) \tag{3.5}$$

We start with a prior density $p(\mathbf{s}_{t-1}|\mathbf{z}_{1:t-1})$ and generates a posterior density $p(\mathbf{s}_t|\mathbf{z}_{1:t})$, under the Bayesian framework, in the following two steps:
**Step:1 (Prediction)** The predicted density $p(\mathbf{s}_t|\mathbf{z}_{1:t-1})$ is calculated by marginalizing the $p(\mathbf{s}_t, \mathbf{s}_{t-1}|\mathbf{z}_{1:t-1})$ over the previous state $\mathbf{s}_{t-1}$:

$$\begin{aligned} p(\mathbf{s}_t|\mathbf{z}_{1:t-1}) &= \int p(\mathbf{s}_t, \mathbf{s}_{t-1}|\mathbf{z}_{1:t-1}) d\mathbf{s}_{t-1} \\ &= \int p(\mathbf{s}_t|\mathbf{s}_{t-1}, \mathbf{z}_{1:t-1}) p(\mathbf{s}_{t-1}|\mathbf{z}_{1:t-1}) d\mathbf{s}_{t-1} \\ &= \int p(\mathbf{s}_t|\mathbf{s}_{t-1}) p(\mathbf{s}_{t-1}|\mathbf{z}_{1:t-1}) d\mathbf{s}_{t-1} \end{aligned} \tag{3.6}$$

The above set of equations is also called Chapman-Kolmogorov equation.
**Step:2 (Update)** Given the prediction density $p(\mathbf{s}_t|\mathbf{z}_{1:t-1})$ and the reception of new measurement $\mathbf{z}_t$ at time $t$, the posterior density $p(\mathbf{s}_t|\mathbf{z}_{1:t})$ is computed by:

$$\begin{aligned} p(\mathbf{s}_t|\mathbf{z}_t, \mathbf{z}_{1:t-1}) &= \frac{p(\mathbf{z}_t, \mathbf{z}_{1:t-1}|\mathbf{s}_t) p(\mathbf{s}_t)}{p(\mathbf{z}_t, \mathbf{z}_{1:t-1})} \\ &= \frac{p(\mathbf{z}_t, \mathbf{z}_{1:t-1}, \mathbf{s}_t)}{p(\mathbf{s}_t)} \frac{p(s_t)}{p(\mathbf{z}_t, \mathbf{z}_{1:t-1})} \\ &= \frac{p(\mathbf{z}_t|\mathbf{z}_{1:t-1}, \mathbf{s}_t) p(\mathbf{z}_{1:t-1}, \mathbf{s}_t)}{p(\mathbf{z}_t, \mathbf{z}_{1:t-1})} \\ &= \frac{p(\mathbf{z}_t|\mathbf{z}_{1:t-1}, \mathbf{s}_t) p(\mathbf{s}_t|\mathbf{z}_{1:t-1}) p(\mathbf{z}_{1:t-1})}{p(\mathbf{z}_t|\mathbf{z}_{1:t-1}) p(\mathbf{z}_{1:t-1})} \\ &= \frac{p(\mathbf{z}_t|\mathbf{s}_t) p(\mathbf{s}_t|\mathbf{z}_{1:t-1})}{p(\mathbf{z}_t|\mathbf{z}_{1:t-1})} \end{aligned} \tag{3.7}$$

The denominator of (3.7) is the normalization constant given by:

$$p(\mathbf{z}_t|\mathbf{z}_{1:t-1}) = \int p(\mathbf{z}_t, \mathbf{s}_t|\mathbf{z}_{1:t-1})d\mathbf{s}_t = \int p(\mathbf{z}_t|\mathbf{s}_t)p(\mathbf{s}_t|\mathbf{z}_{1:t-1})d\mathbf{s}_t \tag{3.8}$$

Using (3.6) and (3.8), (3.7) can be written as:

$$p(\mathbf{s}_t|\mathbf{z}_{1:t}) \propto p(\mathbf{z}_t|\mathbf{s}_t) \int p(\mathbf{s}_t|\mathbf{s}_{t-1})p(\mathbf{s}_{t-1}|\mathbf{z}_{1:t-1})d\mathbf{s}_{t-1} \tag{3.9}$$

The above expression is the recursive formula for posterior density estimation. The details of each term are: i) $p(\mathbf{z}_t|\mathbf{s}_t)$ is called the likelihood and is characterized by the measurement/observation model of dynamic state-space model. It represents the dependence of the observation on the actual state; ii) $p(\mathbf{s}_t|\mathbf{s}_{t-1})$ is called state transition probability and is characterized by the motion model of dynamic state-space model; iii) $p(\mathbf{s}_{t-1}|\mathbf{z}_{1:t-1})$ is called prior knowledge, which is assumed to be known initially and then recursively updated

### 3.1.2 Kalman Filters

The posterior pdf is Gaussian, if motion and measurement models are linear with additive Gaussian noise and Gaussian prior.i.e

$$\begin{aligned} \mathbf{s}_t &= \mathbf{A}_{t-1}\mathbf{s}_{t-1} + \mathbf{v}_{t-1} \\ \mathbf{z}_t &= \mathbf{B}_t\mathbf{s}_t + \mathbf{w}_t \end{aligned} \tag{3.10}$$

where $\mathbf{A}_{t-1}$ and $\mathbf{B}_t$ are system and observation matrices, $\mathbf{v}_{t-1} \approx \mathcal{N}(0, \mathbf{Q}_{t-1})$ and $\mathbf{w}_t \approx \mathcal{N}(0, \mathbf{R}_t)$ are mutually independent modelling and process noise with covariances $\mathbf{Q}_{t-1}$ and $\mathbf{R}_t$ respectively.

Under this assumption, the posterior pdf can be characterized by its first two moments. The Kalman filter [26] gives the closed form solution and an optimal MMSE estimate of posterior. It recursively calculates the mean and covariance of posterior pdf as:

**(Prediction)**

$$\hat{\mathbf{s}}_{t|t-1} = \mathbf{A}_{t-1}\hat{\mathbf{s}}_{t-1|t-1} \tag{3.11}$$

$$\mathbf{P}_{t|t-1} = \mathbf{A}_{t-1}\mathbf{P}_{t-1|t-1}\mathbf{A}_{t-1}^T + \mathbf{Q}_{t-1} \tag{3.12}$$

**(Update)**

$$\tilde{\mathbf{z}}_t = \mathbf{z}_t - \mathbf{B}_t\hat{\mathbf{s}}_{t|t-1} \tag{3.13}$$

$$\mathbf{S}_t = \mathbf{B}_t\mathbf{P}_{t|t-1}\mathbf{B}_t^T + \mathbf{R}_t \tag{3.14}$$

$$\mathbf{K}_t = \mathbf{P}_{t|t-1}\mathbf{B}_t^T\mathbf{S}_t^{-1} \tag{3.15}$$

$$\hat{\mathbf{s}}_{t|t} = \hat{\mathbf{s}}_{t|t-1} + \mathbf{K}_t\tilde{\mathbf{z}}_t \tag{3.16}$$

$$\mathbf{P}_{t|t} = (\mathbf{I} - \mathbf{K}_t\mathbf{B}_t)\mathbf{P}_{t|t-1} \tag{3.17}$$

where $\hat{\mathbf{s}}_{t|t}$ and $\mathbf{P}_{t|t}$ are posterior mean and covariance matrix estimate, $\mathbf{K}_t$ is the Kalman gain weighting matrix, $\tilde{\mathbf{z}}_t$ is the innovation i.e. difference between received

measurement $\mathbf{z}_t$ and predicted measurement $\mathbf{B}_t\hat{\mathbf{s}}_{t|t-1}$ and $\mathbf{S}_t$ is the covariance matrix of innovation.

Kalman filter is further extended to handle non linear state and observation models. e.g. Extended Kalman filter (EKF) [27]: It linearly approximates the nonlinear models by using Taylor series before using the original formulation; Unscented Kalman filter (UKF) [64]: It approximates the posterior distribution through a set of deterministically chosen points. Readers are referred to [29, 59] for further details of Kalman filter equation and its extensions.

### 3.1.3 Particle Filters

Particle filters (PFs) are a family of techniques that use Monte Carlo simulations to find a numerical solution of the posterior density for general non-linear and non-gaussian systems. PF estimates the posterior density $p(\mathbf{s}_t|\mathbf{z}_{1:t})$ at $t$ by using a weighted sum of $N$ discrete random samples (, or particles) drawn from the posterior space, as follows:

$$p(\mathbf{s}_t|\mathbf{z}_{1:t}) \approx \sum_{j=1}^{N} \omega_t^j \delta(\mathbf{s}_t - \mathbf{s}_t^j) \tag{3.18}$$

where $\{\mathbf{s}_t^j, j = 1, \ldots, N\}$ is a set of support points with associated weights $\{\omega_t^j, j = 1, \ldots, N\}$, given as:

$$\omega_t^j = \omega_{t-1}^j \frac{p(\mathbf{z}_t|\mathbf{s}_t^j)p(\mathbf{s}_t^j|\mathbf{s}_{t-1}^j)}{q(\mathbf{s}_t^j|\mathbf{s}_{t-1}^j, \mathbf{z}_t)} \tag{3.19}$$

where $q(\mathbf{s}_t^j|\mathbf{s}_{t-1}^j, \mathbf{z}_t)$ is the proposal distribution.

**Proof:** Let $\mathbf{s}_{1:t}$ and $\mathbf{z}_{1:t}$ represents sequence of all target states and observation up to time $t$, $p(\mathbf{s}_{1:t}|\mathbf{z}_{1:t})$ denotes joint posterior density and $p(\mathbf{s}_t|\mathbf{z}_{1:t})$ be its marginal. Then,$p(\mathbf{s}_{1:t}|\mathbf{z}_{1:t})$ can be approximated as follows:

$$p(\mathbf{s}_{1:t}|\mathbf{z}_{1:t}) \approx \sum_{j=1}^{N} \omega_t^j \delta(\mathbf{s}_{1:t} - \mathbf{s}_{1:t}^j) \tag{3.20}$$

It is difficult to draw samples from the posterior distribution because it is not available, multivariate and non-standard [29]. A proposal distribution or importance density $q(\mathbf{s}_{1:t}|\mathbf{z}_{1:t})$ is commonly used which should resemble the posterior distribution and easy to generate samples. By this representation of samples, it is still possible to approximate the posterior distribution, provided the weights can be defined up to proportionality by the following:

$$\omega_t^j \propto \frac{p(\mathbf{s}_{1:t}^j|\mathbf{z}_{1:t})}{q(\mathbf{s}_{1:t}^j|\mathbf{z}_{1:t})} \tag{3.21}$$

For recursive updation of weights, the importance density is chosen to factorize such that:

$$q(\mathbf{s}_{1:t}|\mathbf{z}_{1:t}) = q(\mathbf{s}_t|\mathbf{s}_{1:t-1}, \mathbf{z}_{1:t})q(\mathbf{s}_{1:t-1}|\mathbf{z}_{1:t-1}) \tag{3.22}$$

This helps to obtain samples $\mathbf{s}_{1:t}^j \sim q(\mathbf{s}_{1:t}|\mathbf{z}_{1:t})$ by combining each of existing samples $\mathbf{s}_{1:t-1}^j \sim q(\mathbf{s}_{1:t-1}|\mathbf{z}_{1:t-1})$ with the new state $\mathbf{s}_t^j \sim q(\mathbf{s}_t|\mathbf{s}_{t-1}, \mathbf{z}_t)$ [29].

For recursive estimation, $p(\mathbf{s}_{1:t}|\mathbf{z}_{1:t})$ can be expressed as:

$$p(\mathbf{s}_{1:t}|\mathbf{z}_{1:t}) = p(\mathbf{s}_{1:t}|\mathbf{z}_t, \mathbf{z}_{1:t-1}) \tag{3.23}$$

$$= \frac{p(\mathbf{s}_{1:t}, \mathbf{z}_t, \mathbf{z}_{1:t-1})}{p(\mathbf{z}_t, \mathbf{z}_{1:t-1})} \tag{3.24}$$

$$= \frac{p(\mathbf{z}_t|\mathbf{s}_{1:t}, \mathbf{z}_{1:t-1})p(\mathbf{s}_t|\mathbf{s}_{1:t-1}, \mathbf{z}_{1:t-1})p(\mathbf{s}_{1:t-1}|\mathbf{z}_{1:t-1})}{p(\mathbf{z}_t|\mathbf{z}_{1:t-1})} \tag{3.25}$$

$$= p(\mathbf{s}_{1:t-1}|\mathbf{z}_{1:t-1}) \frac{p(\mathbf{z}_t|\mathbf{s}_t)p(\mathbf{s}_t|\mathbf{s}_{t-1})}{p(\mathbf{z}_t|\mathbf{z}_{1:t-1})} \tag{3.26}$$

Using (3.26) in (3.22), the weights can be updates as:

$$\omega_t^j \propto \frac{p(\mathbf{s}_{1:t-1}^j|\mathbf{z}_{1:t-1})}{q(\mathbf{s}_{1:t-1}^j|\mathbf{z}_{1:t-1})} \frac{p(\mathbf{z}_t|\mathbf{s}_t^j)p(\mathbf{s}_t^j|\mathbf{s}_{t-1}^j)}{q(\mathbf{s}_t^j|\mathbf{s}_{1:t-1}^j, \mathbf{z}_{1:t})} \tag{3.27}$$

$$\propto \omega_{t-1}^j \frac{p(\mathbf{z}_t|\mathbf{s}_t^j)p(\mathbf{s}_t^j|\mathbf{s}_{t-1}^j)}{q(\mathbf{s}_t^j|\mathbf{s}_{1:t-1}^j, \mathbf{z}_{1:t})} \tag{3.28}$$

With the assumption that the current state of proposal distribution is dependent on the previous state $\mathbf{s}_{t-1}$ and the current mesurement $\mathbf{z}_t$, (3.28) is modified to:

$$\omega_t^j = \omega_{t-1}^j \frac{p(\mathbf{z}_t|\mathbf{s}_t^j)p(\mathbf{s}_t^j|\mathbf{s}_{t-1}^j)}{q(\mathbf{s}_t^j|\mathbf{s}_{t-1}^j, \mathbf{z}_t)} \tag{3.29}$$

□

The main difference of applying PFs for visual targets compared to point targets lies in the design of the state vector. It is not suitable to stack visual target shape and appearance into a long state vector due to computational complexity. Instead, the state vector is divided into two sub-state vectors corresponding to the visual target shape and appearance. Dynamic model is applied to the shape parameters of the state vector and measurement is generated from the appearance of the state vector.

The choice of importance density is very critical in the design of particle filters. The most commonly used importance function is $q(\mathbf{s}_t|\mathbf{s}_{t-1}, \mathbf{z}_t) = p(\mathbf{s}_t|\mathbf{s}_{t-1})$ which is easy to use and simplifies the expressions for weight update, but it is often far from optimal. To alleviate this weakness, Kalman filter or Extended Kalman filter [60] can be used to find an approximation of $q(\mathbf{s}_t|\mathbf{s}_{t-1}, \mathbf{z}_t)$ from which it is easy to generate samples. Moreover, due to a particular choice of importance function of form (3.22), variance of importance weights increases over time [60]. It results in a situation after sometime, where all particles will have negligible weights except one, (called degeneracy phenomenon). One effective remedy is to discard the particles with low importance weights. Particles with high importance weights are redistributed evenly across the posterior. This is called called resampling. It is done when effective sample size $N_{eff} = \frac{1}{\sum_{j=1}^{N}(\omega_t^j)^2}$ is smaller than a threshold. There are a number of resampling

scheme of different complexities e.g. multinomial resampling [61], systematic resampling [65], and residual resampling [66] etc. One critical side effect of resampling is the sample impoverishment. i.e. diversity among the particles decreases due to repeated selection of only those particles which have high weights.

Several attempts have been made to overcome degeneracy and sample impoverishment by different choice of importance density and variation in resampling method. This results in different versions of PFs. e.g. sampling importance resampling (SIR) or bootstrap or CONDENSATION [61], auxiliary sampling importance resampling filter [62], regularized particle filter with improved sample diversity [63], local linearization PF [60, 64], mean shift embedded PF [41] and many more.

## 3.2 Mean Shift Tracking

Mean shift, originally formulated by [67, 68], is a non-parametric kernel-based method to find the local maxima (or mode) of the unknown probability distribution function. It starts with the assumption of the location of mode which is refined by iterative steps towards the true maxima. The mean shift vector is proportional to the normalized density gradient and points towards the direction of maximum increase in density; therefore it is a gradient ascent algorithm. It is capable of solving many different problems in the field of computer vision like segmentation, tracking, nonlinear edge preserving image smoothing etc.

In the context of visual tracking, it was first generalized and analyzed by [1]. The tracking is formulated as maximizing the Bhattacharyya coefficient, or minimizing the Bhattacharyya distance, between the reference and target objects kernel weighted color histogram. The search is performed by seeking the target location in the previous frame and then finding the location in the current frame where the target similarity with the reference is maximized. Being a gradient optimization method, it is easy to implement and computationally fast with good tracking results.

The main drawback of mean shift tracking is that it may drift away or loss of tracking, especially in cases of background clutter with similar colors to the object of interest (as color histogram discards the spatial information of the target), object occlusions, pose changes of large objects, and fast change of object motion.

### 3.2.1 Mean Shift for Finding the Mode of Density

The general form of kernel density estimator for a $d-$dimensional random variable with independent and identically distributed samples $\{\mathbf{x}_i, i = 1, \ldots, n\}$ [69, 70] at a point $\mathbf{x}$ is defined as:

$$\hat{p}_K(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} K_{\mathbf{\Sigma}}(\mathbf{x} - \mathbf{x}_i) = \frac{1}{n}\sum_{i=1}^{n} |\mathbf{\Sigma}|^{\frac{-1}{2}} K(\mathbf{\Sigma}^{-1/2}(\mathbf{x} - \mathbf{x}_i)) \tag{3.30}$$

where $K$ is the $d-$ dimensional kernel, $\mathbf{\Sigma}$ is the $d \times d$ symmetric positive definite bandwidth matrix. For radially symmetric kernel, $K(\mathbf{z}) = c_k k\|\mathbf{z}\|^2$ and $\mathbf{\Sigma} = \sigma^2\mathbf{I}$, the

kernel density estimator becomes:

$$\hat{p}_K(\mathbf{x}) = \frac{1}{n\sigma^d}\sum_{i=1}^{n} K(\frac{\mathbf{x}-\mathbf{x}_i}{\sigma}) = \frac{c_k}{n\sigma^d}\sum_{i=1}^{n} k\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{\sigma}\right\|\right) \tag{3.31}$$

where $k$ is the profile of the kernel and $c_k$ is the normalization constant.

The mode of the estimated density can be obtained by taking the gradient of (3.31) and setting it equal to zero. i.e.

$$\nabla\hat{p}_K(\mathbf{x}) = \frac{2c_k}{n\sigma^{d+2}}\sum_{i=1}^{n}(\mathbf{x}-\mathbf{x}_i)k'\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{\sigma}\right\|^2\right) \tag{3.32}$$

where $k'(\mathbf{z})$ is the first derivative of $k(\mathbf{z})$. By setting $G(\mathbf{z}) = c_g g\|\mathbf{z}\|^2$, the kernel $K$ is called the shadow of the kernel $G$, when the condition $g(\mathbf{z}) = -k'(\mathbf{z})$ is satisfied. Substituting the shadow kernel in (3.32) and denoting $\hat{p}_G(\mathbf{x}) = \frac{c_g}{n\sigma^d}\sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{\sigma}\right\|\right)$, it follows that:

$$\nabla\hat{p}_K(\mathbf{x}) = \frac{2}{\sigma^2}\frac{c_k}{n\sigma^d}\sum_{i=1}^{n}\mathbf{x}_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{\sigma}\right\|\right) - \frac{2\mathbf{x}c_k}{\sigma^2 c_g}\hat{p}_G(\mathbf{x}) \tag{3.33}$$

After some manipulation, (3.33) reduces to:

$$\frac{1}{2}\sigma^2 c\frac{\nabla\hat{p}_K(\mathbf{x})}{\hat{p}_G(\mathbf{x})} = \frac{\sum_{i=1}^{n}\mathbf{x}_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{\sigma}\right\|\right)}{\sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{\sigma}\right\|\right)} - \mathbf{x} \tag{3.34}$$

where $c = \frac{c_g}{c_k}$ is a constant. The right hand side of the above equation is called the isotropic mean shift and is denoted as:

$$m_G(\mathbf{x}) = \frac{\sum_{i=1}^{n}\mathbf{x}_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{\sigma}\right\|\right)}{\sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{\sigma}\right\|\right)} - \mathbf{x} \tag{3.35}$$

The magnitude of mean shift vector decreases to guarantee the convergence and its direction points to the nearest stationary point (local mode) of the density function estimate. Moreover, mean shift computed with kernel $G$ is proportional to normalized density gradient obtained with kernel $K$, thus it can be considered as gradient ascent.

### 3.2.2 Mean Shift for Target Localization

This section gives a brief review of the isotropic and anisotropic mean shift for visual object tracking. The details mainly follow [1, 40].

#### Isotropic Mean Shift For Target Localization

The target candidate at location $\mathbf{y}$ is characterized by radially symmetric kernel weighted intensity (, or color) histogram as:

$$\hat{p}(\mathbf{y}) = \{\hat{p}_u(\mathbf{y})\}_{u=1\ldots m}, \qquad \hat{p}_u(\mathbf{y}) = C_h\sum_{i=1}^{n_\sigma} k(\left\|\frac{\mathbf{y}-\mathbf{x}_i}{\sigma}\right\|^2)\delta[b(\mathbf{x}_i)-u] \tag{3.36}$$

where $\hat{p}(\mathbf{y})$ is the pdf estimate of the candidate, $\hat{p}_u(\mathbf{y})$ is $u^{th}$ bin of the histogram, $m$ is the total number of bins, $b(\mathbf{x}_i)$ is a function that associates the pixel at the location $\mathbf{x}_i$, the index of its bin in the quantized feature space, $\mathbf{x}_i$ is summed over all pixels within the object region, $\sigma$ is the kernel bandwidth and $C_h$ is the normalization constant. Similarly, pdf estimate of the reference target model can be obtained $\hat{q} = \{\hat{q}_u\}_{u=1\ldots m}$.

Target position is localized in the current frame by searching in the neighborhood of location $\mathbf{y}$, with the aim to maximize the Bhattacharyya similarity between target model and candidate pdf. The distance measure is defined as:

$$d(\mathbf{y}) = \sqrt{1 - \rho[\hat{p}(\mathbf{y}), \hat{q}]} \tag{3.37}$$

where

$$\rho[\hat{p}(\mathbf{y}), \hat{q}] = \sum_{u=1}^{m} \sqrt{\hat{p}(\mathbf{y}), \hat{q}} \tag{3.38}$$

where $\rho$ is the Bhattacharyya similarity coefficient.
Let the initial guess for the location of the maximum of $\rho$ be $\mathbf{y}_0$. Using the first-order Taylor series expansion of the Bhattacharyya coefficient (3.38) around the values $\hat{p}(\mathbf{y}_0)$:

$$\rho[\hat{p}(\mathbf{y}), \hat{q}] \approx \sum_{u=1}^{m} \sqrt{\hat{p_u}(\mathbf{y}_0), \hat{q}_u} + \frac{1}{2} \sum_{i=1}^{m} (\hat{p_u}(\mathbf{y}) - \hat{p_u}(\mathbf{y}_0)) \sqrt{\frac{\hat{q}_u}{\hat{p_u}(\mathbf{y}_0)}} \tag{3.39}$$

$$\approx \frac{1}{2} \sum_{u=1}^{m} \sqrt{\hat{p_u}(\mathbf{y}_0), \hat{q}_u} + \frac{1}{2} \sum_{i=1}^{m} \hat{p_u}(\mathbf{y}) \sqrt{\frac{\hat{q}_u}{\hat{p_u}(\mathbf{y}_0)}} \tag{3.40}$$

Using (3.36), we have:

$$\rho[\hat{p}(\mathbf{y}), \hat{q}] = \frac{1}{2} \sum_{u=1}^{m} \sqrt{\hat{p_u}(\mathbf{y}_0), \hat{q}_u} + \frac{C_h}{2} \sum_{i=1}^{n} \omega_i k \left( \left\| \frac{y - x_i}{\sigma} \right\|^2 \right) \tag{3.41}$$

where

$$\omega_i = \sum_{u=1}^{m} \sqrt{\frac{\hat{q}_u}{\hat{p_u}(\mathbf{y}_0)}} \delta[b(\mathbf{x}_i) - u] \tag{3.42}$$

To minimize the distance (3.37), we need to maximize $\rho$ in (3.41). It is done by taking the gradient of (3.41) and setting it to zero which results in the following expression for MS position update:

$$\hat{\mathbf{y}}_1 = \frac{\sum_{i=1}^{n} \mathbf{x}_i \omega_i g \left( \left\| \frac{\mathbf{y}_0 - \mathbf{x}_i}{\sigma} \right\|^2 \right)}{\sum_{i=1}^{n} \omega_i g \left( \left\| \frac{\mathbf{y}_0 - \mathbf{x}_i}{\sigma} \right\|^2 \right)} \tag{3.43}$$

Type of kernel (i.e. isotropic or non-isotropic) and its bandwidth parameter estimation are essential issues in MS tracking. The original formulation, proposed by [1], uses isotropic kernel having convex and monotonic decreasing kernel profile assigning smaller weight to the pixels further from the center, while the selection of the

bandwidth related to the scale of the target is a design parameter whose value is empirically determined. The bandwidth calculation is made adaptive by starting from the empirical value and checking in $\pm 10\%$ variation. It gives a very crude estimate of target scale, which is of little significance for practical scenario, where the target may undergo affine transformation during the course of tracking. We now describe the extension of isotropic MS by [40] that uses anisotropic kernel, with adaptive selection of the bandwidth by MS iterations.

**Anisotropic Mean Shift for Target Localization**

The target candidate at location $\mathbf{y}$ is characterized by anisotropic kernel weighted intensity (, or color) histogram as:

$$p_u(\mathbf{y}, \mathbf{\Sigma}) = \frac{c}{|\mathbf{\Sigma}|^{\frac{1}{2}}} \sum_j k(\tilde{\mathbf{y}_j}^T \mathbf{\Sigma}^{-1} \tilde{\mathbf{y}_j}) \delta[b_u(I(\mathbf{y}_j)) - u] \tag{3.44}$$

where $\tilde{\mathbf{y}}_j = (\mathbf{x}_j - \mathbf{y})$, $\mathbf{\Sigma}$ is an anisotropic kernel bandwidth matrix, $b_u(I(\mathbf{y}_j))$ is the index of color histogram bin at location $\mathbf{y}_j$ associated with the target candidate , $\mathbf{y}_j$ is summed over all pixels within the bounding box, $c$ is a constant used for normalization, $u = 1, \cdots, m$, m is the total number of bins, $k(\cdot)$ is a spatial kernel profile. Similarly, pdf estimate of the reference target model can be obtained.

Similar to isotropic MS, the expressions for Bhattacharyya similarity and position update can be expressed as:

$$\rho \approx \sum_u \frac{1}{2} \sqrt{q_u p_u(\mathbf{y}_0, \mathbf{\Sigma}_0)} + \frac{c}{2|\mathbf{\Sigma}|^{\frac{1}{2}}} \sum_j \omega_j k(\tilde{\mathbf{y}}_j^T \mathbf{\Sigma}^{-1} \tilde{\mathbf{y}}_j) \tag{3.45}$$

$$\omega_j = \sum_u \sqrt{\frac{q_u}{p_u(\mathbf{y}_0, \mathbf{\Sigma}_0)}} \delta[b_u(I(\mathbf{y}_j)) - u] \tag{3.46}$$

$$\hat{y} = \frac{\sum_{j=1}^n g(\tilde{\mathbf{y}}_j^T \mathbf{\Sigma}^{-1} \tilde{\mathbf{y}}_j) \omega_j \mathbf{x}_j}{\sum_{j=1}^n g(\tilde{\mathbf{y}}_j^T \mathbf{\Sigma}^{-1} \tilde{\mathbf{y}}_j) \omega_j} \tag{3.47}$$

**Proof:**

$$\frac{\partial}{\partial y} \left( \frac{c}{2|\mathbf{\Sigma}|^{\frac{1}{2}}} \sum_j \omega_j k(\tilde{\mathbf{y}}_j^T \mathbf{\Sigma}^{-1} \tilde{\mathbf{y}}_j) \right) = 0 \tag{3.48}$$

$$\frac{c}{2|\mathbf{\Sigma}|^{\frac{1}{2}}} \sum_j \omega_j (-k'(\tilde{\mathbf{y}}_j^T \mathbf{\Sigma}^{-1} \tilde{\mathbf{y}}_j))(2\tilde{\mathbf{y}} \mathbf{\Sigma}^{-1}) = 0 \tag{3.49}$$

After some manipulations and using $g(.) = -k'(.)$ and $\tilde{\mathbf{y}}_j = (\mathbf{x}_j - \mathbf{y})$ in (3.49), we get

$$\hat{y} = \frac{\sum_{j=1}^n g(\tilde{\mathbf{y}}_j^T \mathbf{\Sigma}^{-1} \tilde{\mathbf{y}}_j) \omega_j \mathbf{x}_j}{\sum_{j=1}^n g(\tilde{\mathbf{y}}_j^T \mathbf{\Sigma}^{-1} \tilde{\mathbf{y}}_j) \omega_j} \tag{3.50}$$

□

To estimate $\hat{\mathbf{\Sigma}}$, a $\gamma$-normalized kernel bandwidth $\mathbf{\Sigma}$ is applied to $\rho$ (similar to [71]). The kernel bandwidth matrix estimate is obtained by setting $\nabla_{\mathbf{\Sigma}}(|\mathbf{\Sigma}|^{\gamma/2}\rho(p,q)) = 0$, yielding,

$$\hat{\mathbf{\Sigma}} = \frac{2}{1-\gamma}\frac{\sum_{j=1}^{n}\omega_j\ g(\tilde{\mathbf{y}}_j^T\mathbf{\Sigma}^{-1}\tilde{\mathbf{y}}_j)\tilde{\mathbf{y}}_j\tilde{\mathbf{y}}_j^T}{\sum_{j=1}^{n}\omega_j\ k(\tilde{\mathbf{y}}_j^T\mathbf{\Sigma}^{-1}\tilde{\mathbf{y}}_j)} \tag{3.51}$$

where $\gamma$ is empirically determined, and $\tilde{\mathbf{y}}_j = (\mathbf{x}_j - \hat{\mathbf{y}})$.
**Proof:**

$$\frac{\partial}{\partial\mathbf{\Sigma}}\left(|\mathbf{\Sigma}|^{\frac{\gamma}{2}}\frac{c}{2|\mathbf{\Sigma}|^{\frac{1}{2}}}\sum_j\omega_j k(\tilde{\mathbf{y}}_j^T\mathbf{\Sigma}^{-1}\tilde{\mathbf{y}}_j)\right) = 0 \tag{3.52}$$

$$\frac{\partial}{\partial\mathbf{\Sigma}}\left(|\mathbf{\Sigma}|^{\frac{\gamma-1}{2}}\sum_j\omega_j k(\tilde{\mathbf{y}}_j^T\mathbf{\Sigma}^{-1}\tilde{\mathbf{y}}_j)\right) = 0 \tag{3.53}$$

Using matrix differentiation formula $\frac{d}{dA}|A| = |A|A^{-1}$ and representing $k(\tilde{\mathbf{y}}_j^T\mathbf{\Sigma}^{-1}\tilde{\mathbf{y}}_j) = k(\cdot)$, we get:

$$|\mathbf{\Sigma}|^{\frac{\gamma-1}{2}}\left(\sum_j\omega_j(k'(\cdot)(-\tilde{\mathbf{y}}_j\tilde{\mathbf{y}}_j^T\mathbf{\Sigma}^{-2})\right) + \frac{\gamma-1}{2}|\mathbf{\Sigma}|^{\frac{\gamma-1}{2}-1}|\mathbf{\Sigma}|\mathbf{\Sigma}^{-1}\left(\sum_j\omega_j k(\cdot)\right) = 0$$

Multiplying both sides by $\mathbf{\Sigma}^2$:

$$|\mathbf{\Sigma}|^{\frac{\gamma-1}{2}}\left(\sum_j\omega_j(-k'(\cdot)(-\tilde{\mathbf{y}}_j\tilde{\mathbf{y}}_j^T)\right) - \frac{1-\gamma}{2}|\mathbf{\Sigma}|^{\frac{\gamma-1}{2}}\mathbf{\Sigma}\left(\sum_j\omega_j k(\cdot)\right) = 0$$

After some algebraic manipulations and using $g(.) = -k'(.)$, we get:

$$|\mathbf{\Sigma}|^{\frac{\gamma-1}{2}}\frac{1-\gamma}{2}\left(\frac{2}{1-\gamma}\sum_j\omega_j g(\cdot)\tilde{\mathbf{y}}_j\tilde{\mathbf{y}}_j^T - \mathbf{\Sigma}\sum_j\omega_j k(\cdot)\right) = 0$$

This finally results:

$$\hat{\mathbf{\Sigma}} = \frac{2}{1-\gamma}\frac{\sum_{j=1}^{n}\omega_j\ g(\tilde{\mathbf{y}}_j^T\mathbf{\Sigma}^{-1}\tilde{\mathbf{y}}_j)\tilde{\mathbf{y}}_j\tilde{\mathbf{y}}_j^T}{\sum_{j=1}^{n}\omega_j\ k(\tilde{\mathbf{y}}_j^T\mathbf{\Sigma}^{-1}\tilde{\mathbf{y}}_j)} \tag{3.54}$$

□

The estimation of (3.47) and (3.51) are done alternatively in each iteration. The iterations continue until the estimated parameters converge, or a pre-specified maximum number of iterations are reached. The parameters of the bounding box (width, height and orientation) are calculated by the eigen decomposition of the bandwidth matrix.

## 3.3 Tracking with Local Features

Local features extraction and matching have a lot of application in the field of computer vision like detection, tracking, panorama stretching and many more. These features are based on the local properties of group of neighboring pixels of an image at a particular interest point like edges/corners/texture etc. These features should be highly distinctive with the following properties:

1. Easy to detect/locate with well defined position in the image space.
2. Tolerant to image noise, intensity variation, rotation and scaling, minor changes in viewing direction etc
3. Robust and easy to match against large data base of local feature.

In the context of tracking, local features in the target region are extracted and compared with the feature set of the entire image area of consecutive frames. This establishes correspondence which provide a reliable and robust estimation of transformation of the bounding polygon around target. Moreover, to mitigate the effect of noise and incorrect matches, outliers and inliers are separated using a random sample consensus (RANSAC) and transformation is estimated using inliers only.

These features, being local in nature gives good result in case of partial occlusion, camera jitter etc. However, the main drawback is that if the number of good matches falls below the minimum number required to estimate the transformation, the tracking will collapse. This section describes the exiting methodologies for extraction of local features with emphasis on SIFT, transformation estimation and RANSAC

## 3.4 Methods for Extracting Local Features

### 3.4.1 Harris Corner Detector

Harris corner detector [11], identifies the interest points in an image by evaluating the response function $R$, defined as:

$$R = \det(\mathbf{H}) - \kappa(trace(\mathbf{H}))^2 \tag{3.55}$$

where $\det(\mathbf{H}) = \lambda_1\lambda_2$; $trace(H) = \lambda_1 + \lambda_2$; $\lambda_1$ and $\lambda_2$ are the eigen values of $\mathbf{H}$ ; $\kappa$ is the empirically determined constant to give the results in the range $[.04, .06]$ and $\mathbf{H}$ is a $2 \times 2$ autocorrelation/Hessian matrix computed from image derivative as follows:

$$\mathbf{H} = w(x,y) \begin{bmatrix} \sum_{(x,y)\epsilon W} I_x^2 & \sum_{(x,y)\epsilon W} I_x I_y \\ \sum_{(x,y)\epsilon W} I_x I_y & \sum_{(x,y)\epsilon W} I_y^2 \end{bmatrix} \tag{3.56}$$

where $I_x$ and $I_y$ are the derivatives (of pixel intensity) in the x and y direction at point $x, y$ of the image window $W$, $w(x, y)$ is the weighting function which can be uniform or more typically isotropic, circular Gaussian (i.e weighting center values of window more heavily compared to edges).
The criterion to evaluate the response function $R$ is:

1. Corners, if $R > 0$ with large magnitude
2. Edges, if $R < 0$ with large magnitude
3. Flat region, if $|R|$ is small

Harris corner detector is invariant to rotation, affine intensity change but variant to scaling change as shown below [72]:

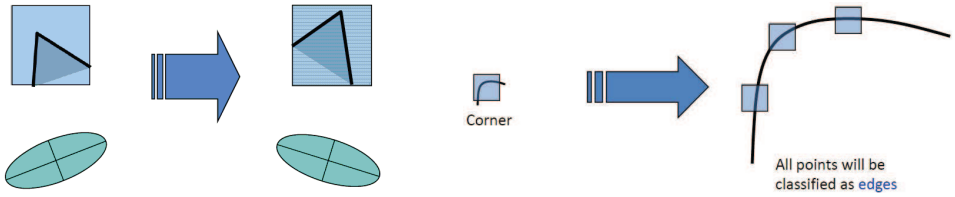


Figure 3.1: Variance/invariance property of Harris corner detector. Left: $H$ visualized by ellipse rotates, but its shape (i.e. eigen values) remains the same; Right: corners are classified as edges after scale change. The figure is taken from [72].

### 3.4.2 Scale-invariant Feature Transform (SIFT)

SIFT [13] gives highly distinctive features that are invariant to affine transformation (translation, 2D rotation and scale) and linear brightness changes. It is widely used in applications that range from robotic mapping and navigation to image recognition, video tracking and image stitching etc. The major steps in the extraction and matching of SIFT features are:

#### SIFT Key Point Localization

A keypoint is a location in the source image with associated scale and orientation calculated by scale space extrema detection and image gradient direction. First, scale space is constructed by convolving the image with Gaussian filters at different scales.

$$L(x, y, \sigma) = G(x, y, \sigma) \star I(x, y) \tag{3.57}$$

where

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \exp(\frac{-(x^2 + y^2)}{\sigma^2}) \tag{3.58}$$

The convolved images are organized into octave of $s + 1$ images. Each octave has an increase of the scale parameter $\sigma$ by a factor of $k$, in between the images, with a maximum of doubling the value of $\sigma$ for the last member of octave. Moreover the images in the next octave are subsampled and stored at a half resolution of previous octave. Difference-of-Gaussian (an approximation of scale normalized Laplacian of

Gaussian (LOG) $\sigma^2\nabla^2 G$) is generated by taking the difference of adjacent blurred images in each octave.

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \tag{3.59}$$

The candidate keypoints are obtained by finding the maxima/minima of difference-of-Gaussian (DOG) pyramid. It is done by comparing the value of each pixel of DOG image with 26-neighborhood pixels (8-surrounding pixels in the same scale and 9 pixels in adjacent scale). The candidate keypoints are further refined by: i) Interpolation of

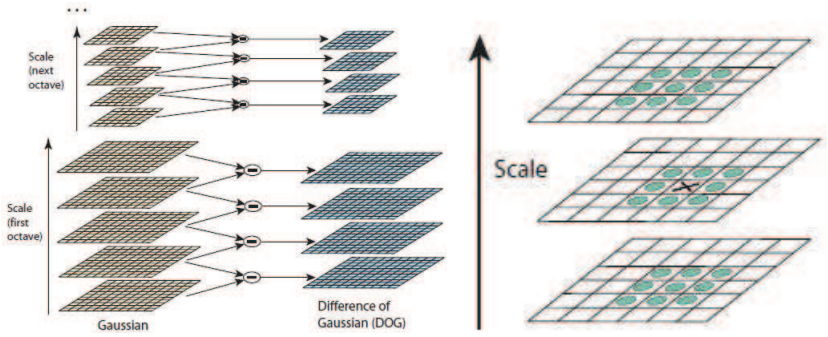


Figure 3.2: Left: scale space images to produce difference-of-Gaussian images; Right: Maxima and minima of difference-of-Gaussian images by comparing a pixel to its 26 neighbours in $3 \times 3$ regions at the current and adjacent scale. The figure is taken from [13].

near-by data through 3D quadratic least square fit; ii) Pruning the extrema that are weak or that corresponds to edges. This is done by evaluating the Hessian matrix at the location and scale of interest point and checking the following condition: $\frac{Tr(H)^2}{\det(H)} < \frac{(r+1)^2}{r}$ where $r$ is the ratio between largest and smallest eigen value of $H$, and is normally chosen to be 10.

Finally the candidate keypoints are assigned orientations as follows: i) A gradient magnitude and orientation is calculated in the neighborhood of the interest points; ii) A 36-bin orientation histogram is then generated. The contribution to the bin is equal to the product of gradient magnitude and Gaussian weight with a $\sigma$ 1.5 times the scale of the keypoint; iii) Orientations corresponding to 80% of maximum peak in the histogram is assigned to the interest points. Thus, a new keypoint is generated having the same position and scale but different orientation.

Finally to assign the orientation to the candidate keypoints, gradient magnitude and orientation is calculated in the neighborhood of the interest point. A 36-bin orientation histogram is then generated. The contribution to the bin is equal to the product of gradient magnitude and Gaussian weight with a $\sigma$ 1.5 times the scale of the keypoint. Orientation corresponding to 80% of maximum peak in the histogram is assigned to the interest point. Thus, a new keypoint is generated having the same position and scale but different orientation.

**SIFT Key Point Description**

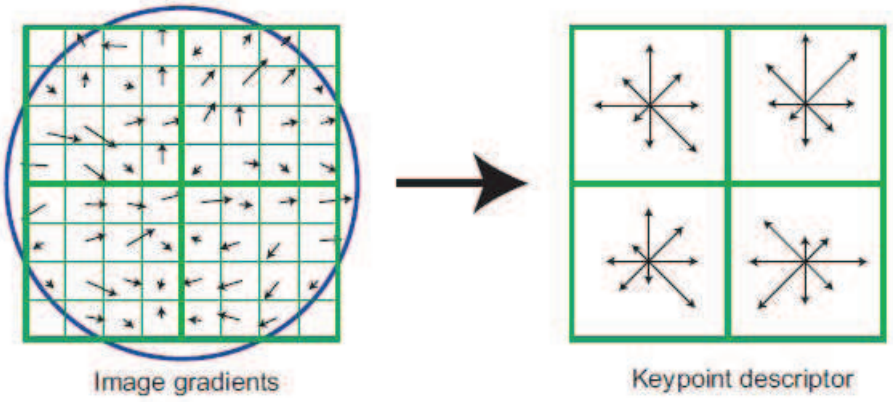


Figure 3.3: A keypoint descriptor computed from an $8 \times 8$ set of samples. The figure is taken from [13].

It is a vector of fixed length and represents the image patch centered at the keypoint as a set of orientation histogram on $4 \times 4$pixel neighborhood. It is computed by calculating the gradient orientation relative to the keypoint orientation in $16 \times 16$ pixel patch of Gaussian image closest in scale to the keypoint scale. Then a 8-bin orientation histogram is generated for each $4 \times 4$ pixel block leading to SIFT feature vector with $4 \times 4 \times 8 = 128$ elements. The contribution of each pixel to bin is weighted by gradient magnitude and gaussian weight with a $\sigma$ 1.5 times the scale of the keypoint.

The descriptor vector is normalized to unit length and thresholded to reduce the effect of illumination. The choice of number of bins and particular values for thresholding are the results of extensive testing proposed by [13]. These parameters can be tuned to get a different number of SIFT features.

**SIFT Key Point Matching**

The SIFT descriptors in two different data set are matched by finding the neighbors with minimum Euclidean distance. A match is selected if the ratio of nearest neighbor Euclidean distance to second neighbor Euclidean distance is less than some predefined threshold. This makes the matching robust to false positive. Fig.3.4, shows the example of above described SIFT features extraction and matching process.

### 3.4.3 SURF: Speeded Up Robust Transform

It is a high speed version of SIFT algorithm, proposed by [14]. It uses the concept of integral images and box filter to detect the features from the determinant of Hessian

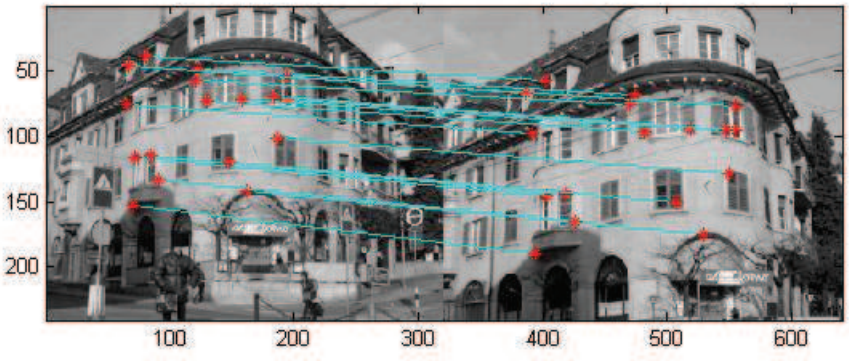


Figure 3.4: SIFT features matching process. Cyan lines shows the matched features (red dots) in two different frames.

matrix instead of LOG, defines as:

$$H = \begin{vmatrix} L_{xx}(x,\sigma) & L_{xy}(x,\sigma) \\ L_{xy}(x,\sigma) & L_{yy}(x,\sigma) \end{vmatrix} \tag{3.60}$$

where $L_{x,\sigma}$ is the convolution of the image $I$ with Gaussian $2^{nd}$ order derivative with scale $\sigma$, approximated by box filter. The interest points are calculated by finding the maxima of above determinant in 3D space $x, y, \sigma$.

The SURF descriptor is calculated by finding the derivatives at 25 sample points using box filter in $4 \times 4$ subregion around the interest point resulting in 4-elements feature vector in each subregion $v = (\sum dx, \sum dy, \sum |dx|, \sum |dy|)$ i.e. $4 \times 4 \times 4 =$ 64element feature vector descriptor.

## 3.5 Transformation Estimation

To estimate the bounding polygon around target, geometric transformation model is estimated from consensus feature points between two frames. This section briefly describes commonly used transformation model and its estimation from the data, in the context of tracking.

### 3.5.1 Affine Transformation Model

Affine transformation of point $(x, y)$ to $(\tilde{x}, \tilde{y})$ is described as follows:

$$\begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \tag{3.61}$$

Given, $n$ consensus feature points, the affine transformation model can be found by solving the following linear equations:

$$\begin{bmatrix} x_1 & y_1 & 0 & 0 & 1 & 0 \\ 0 & 0 & x_1 & y_1 & 0 & 1 \\ \vdots & & & & & \\ x_n & y_n & 0 & 0 & 1 & 0 \\ 0 & 0 & x_n & y_n & 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ t_x \\ t_y \end{bmatrix} = \begin{bmatrix} \tilde{x}_1 \\ \tilde{y}_1 \\ \vdots \\ \tilde{x}_n \\ \tilde{y}_n \end{bmatrix} \tag{3.62}$$

Since the model has 6 unknowns, a minimum of 3 consensus points are needed to estimate model parameters.

### 3.5.2 Similarity Transformation Model

The similarity transformation of point $(x, y)$ to $(\tilde{x}, \tilde{y})$ is of the form:

$$\begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} = \sigma \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \tag{3.63}$$

The linear system of equations $n$ consensus feature points can be written as:

$$\begin{bmatrix} x_1 & -y_1 & 1 & 0 \\ y_1 & x_1 & 0 & 1 \\ \vdots & & & \\ x_n & -y_n & 1 & 0 \\ y_n & x_n & 0 & 1 \end{bmatrix} \begin{bmatrix} \sigma\cos\theta \\ \sigma\sin\theta \\ t_x \\ t_y \end{bmatrix} = \begin{bmatrix} \tilde{x}_1 \\ \tilde{y}_1 \\ \vdots \\ \tilde{x}_n \\ \tilde{y}_n \end{bmatrix} \tag{3.64}$$

Thus this model requires at least 2 consensus points to estimate parameters.

### 3.5.3 Projective Transformation Model

Projective transformation of point $(x, y)$ to $(\tilde{x}, \tilde{y})$ is described by homogeneous coordinates as follows [77]:

$$\begin{bmatrix} \tilde{x} \\ \tilde{y} \\ 1 \end{bmatrix} = \lambda \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{3.65}$$

$\lambda$ is a scaling factor. By performing the multiplication and substituting the value of $\lambda$, this parameter can be eliminated. i.e.

$$\tilde{x}(gx + hy + i) = ax + by + c \tag{3.66}$$

$$\tilde{y}(gx + hy + i) = dx + ey + f \tag{3.67}$$

$$\tag{3.68}$$

The linear system of equations to find 9 unknown parameters of model is described as follows:

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1\tilde{x}_1 & -y_1\tilde{x}_1 & -\tilde{x}_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1\tilde{y}_1 & -y_1\tilde{y}_1 & -\tilde{y}_1 \\ \vdots & & & & & & & & \\ x_n & y_n & 1 & 0 & 0 & 0 & -x_n\tilde{x}_n & -y_n\tilde{x}_n & -\tilde{x}_n \\ 0 & 0 & 0 & x_n & y_n & 1 & -x_n\tilde{y}_n & -y_n\tilde{y}_n & -\tilde{y}_n \end{bmatrix} \begin{bmatrix} a \\ b \\ \vdots \\ i \end{bmatrix} = 0 \tag{3.69}$$

### 3.5.4 RANdom SAmple Consensus (RANSAC)

The usual method of fitting a model to noisy or redundant data is least-squares optimization. It uses all available data to smooth over errors and is a maximum likelihood estimator for Gaussian noise. However, in the presence of gross error (due to the possibility of false matches), least-squares estimation may lead to poor results. In the contest of tracking, Random sample consensus or RANSAC is commonly used which attempts to estimate parameters of transformation model by detecting outliers and excluding them from solution. It is a 2 step probabilistic method to estimate the model parameters, detailed as follows:
In the first step, it starts to estimate model parameters by a randomly selected minimal set of points. The goodness of the estimate is measured by finding the number of data items that fit the estimated model parameters. This is done by calculating the error/cost function for each data point and comparing with the user given threshold ($d_{th}$). It classifies the data by assuming that inliers will be close to the fitted model and outliers not. The process is repeated so that the probability of finding a minimal set of data without outliers can be arbitrarily high.
In the second step, the model model parameters are estimated by least square fit to inliers only. The relationship between the probability $p$ of finding inliers and inlier is [42]:

$$i_{max} = \frac{\log(1-p)}{\log(1-r^n)} \tag{3.70}$$

where $r$ is the ratio of inliers to the total number of features and $n$ is the number of points in the minimal set. For example, for $n = 3$ and $r = 0.6$, $i_{max} = 18$ for 99% confidence level. Readers are referred to [42] discusses different ways to choose $i_{max}$ and $d_{th}$.

CHAPTER 4

# Proposed Online Learning and Video Object Tracking

This chapter gives a general introduction of the proposed appearance online learning on vector space and manifolds. First, online learning of the reference object distributions is described in the context of hybrid tracking methods combining mean shift with local point features correspondences and Bayesian tracking. Finally, online learning and object tracking scheme is discussed that exploits the geometrical structure of manifold under a Bayesian framework. In this thesis, manifolds of interest are symmetric manifolds and Grassmann manifolds.

## 4.1 Online Learning of Local and Global Appearance Features

The basic idea behind our proposed online learning method is to only update the reference model at those frames when they indicate local highest performance of correct tracking without the interference from the background or other objects. e.g. occlusions, intersections and many more. Further, the frequency of update do not need to be high and can be performed in a fixed frame interval, since object changes are usually gradual due to the mechanical movement (e.g. a person takes off an overcoat during the walking). To achieve this, we seek the highest and stable local tracking performance in each fixed interval, to decide whether or not the reference object distribution shall be updated in this interval.

The candidate (or the reference) object appearance in a video frame is described by the spatial-kernel weighted color histograms $p = \{p_u\}$ (or $q = \{q_u\}$), as follows:

$$p_u = \frac{c}{|\Sigma|^{\frac{1}{2}}} \sum_{j=1}^{n} k(\tilde{y_j}^T \Sigma^{-1} \tilde{y_j}) \delta[b_u(I(y_j)) - u] \tag{4.1}$$

where $u = 1, \cdots, m$, $m$ is the total number of bins, $\tilde{y_j} = (y_j - y)$, $\tilde{x_j} = (x_j - x_0)$,

$\Sigma$ is the kernel bandwidth matrix, $b_u(I(y_j))$ is the bin index of color histogram of a candidate object image centered at $y_j$, $c$ is a constant for the normalization, $k$ is the spatial kernel profile, $y$ is the center of the kernel or bounding box, and $I(y_j)$ is the candidate object image within the bounding box. The appearance similarity between a candidate and the reference object is described by the Bhattacharyya coefficient $\rho$ defined as, $\rho(p,q) = \sum_u \sqrt{p_u(y,\Sigma)q_u}$, where $u$ is the histogram bin.

Let $\rho_t = \sum_u \sqrt{q_u^{j-1} p_u^t}$ be the Bhattacharyya coefficient between the current tracked object from the final tracker and the reference object in the previous $(j-1)$th interval, and $\mathbf{x}_{t,i}$ be the 4 corners in the tracked regions $V_t^{(obj)}$. Noting that $q_u^{j-1}$ implies $q_u^t$ for $t \in [(j-2)S+1, (j-1)S]$, where $S$ is the total frames in the interval. Then, the reference model in the $j$th interval is updated if the following two conditions are both satisfied for all frames within the $j$th interval:

$$\begin{cases} \text{dist}_t = \sum_{i=1}^{4} \|\mathbf{x}_{t,i} - \mathbf{x}_{t-1,i}\|^2 < T_1, \\ \rho_t > T_2 \end{cases} \tag{4.2}$$

where $t \in [(j-1)S+1, jS]$, $T_1^{(2)}$ and $T_2^{(2)}$ are empirically selected thresholds. $j = 1, 2, \cdots$, then the best stable high performance frame number is chosen from

$$t^* = \text{argmax}_{t \in [(j-1)S+1, jS]} \ \rho_t \tag{4.3}$$

and the reference object distribution is updated by:

$$q^j = \kappa p^{t^*} + (1-\kappa) q^{j-1} \tag{4.4}$$

where $q^j$ (or, $q^{j-1}$) is the updated reference object pdf in the $j$th (or $(j-1)$th) interval, $\kappa$ is the constant controlling the learning rate ($\kappa = 0.1$ in our tests), $p^{t^*}$ is the pdf where $t^*$ is chosen from (4.3). If (4.2) is not satisfied, then the reference object distribution is not updated, i.e., $q^j \leftarrow q^{j-1}$.

### 4.1.1 Dynamic maintenance of foreground and background feature points

The proposed approach for online dynamic maintenance and updating of consensus points is similar in a spirit to work in [75, 78], however, major change is to combine these methods with global appearance tracking. The method is described as follows:

- The Bhattacharyya similarity coefficient of object region $\rho_t^{(1)} = \sum_u \sqrt{p_u^{t,(1)}(y,\Sigma)\ q_u^t}$ is calculated
- If $\rho_t^{(1)} > T_F$, where $T_F$ is a threshold determined empirically, it indicates that the object area is unlikely to be occluded or newly introduced. The point features in the object area are assigned a score as follows:

$$S_t = \begin{cases} S_{t-1} + 2 & \text{matched consensus point} \\ S_{t-1} - 1 & \text{matched outlier point} \\ median(S_{t-1}) & \text{not matched point} \end{cases} \tag{4.5}$$

  For inliers i.e. point correspondences that fit the transformation, scores are increased. For outliers i.e. point correspondences that do not fit the transformation, scores are decreased. For newly local point features that are only within the candidate object region and do not correspond to any background point features, their score is set to the median value of the previous scores.

- If $\rho_t^{(1)} < T_F$, it indicates that the candidate region likely contains newly introduced area e.g. from partial occlusions or object intersections. Hence the dynamic maintenance is temporarily frozen.

Points in the consensus correspondence set are then sorted according to their scores. A pruning process is applied to remove some feature points with small weights in order to maintain a reasonable size for the points set. This is done by keeping the first $L_F$ ($L_F$=1000 in our tests) highest weight points in the set. New points are added (e.g. due to change pose change or deformation of object) if they fit to the estimated motion model.

In each frame, all feature points in the background region are added into the background set. Feature points in the set are sorted according to their aging. If the total number of feature points exceeds $L_B$ ($L_B$=1500 in our tests), then only the newest $L_B$ feature points are kept while the remaining old aging feature points are removed.

### 4.1.2 A Hybrid Method Combining Adaptive Appearance with Bayesian Tracking

The block diagram of online learning of the reference object appearance distribution with proposed hybrid scheme - 1 is shown in Fig.4.1. In the PF (the top block), the

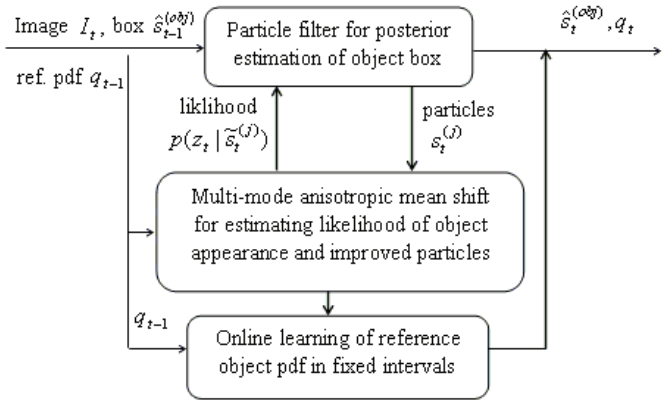


Figure 4.1: Block diagram of hybrid tracking scheme - 1, where online learning of appearance dynamics is added.

state vector $\hat{s}_{t-1}$ describes the shape of a parametric rectangular candidate object box. The initial particles $s_t^{(j)}$ are generated according to the Brownian motion model. Then, the likelihood $p(z_t|\tilde{s}_t^{(j)})$ is computed using the Bhattacharyya distance that is obtained from the dynamic appearance allocated by the anisotropic MS (the middle block). In addition, the rectangular bounding box is partitioned into several disjoint concentric areas. This allows the MS to seek multiple modes that give better descriptions of the spatially-dependent distribution of object appearance. Based on the MS estimates, particles are re-distributed to positions related to large weights, and

posterior pdf of the state vector is then estimated by the PF using the appearance-related likelihood. The online learning (the bottom block) keeps the reference model updated. Moreover, adding online learning in this hybrid tracker has led to more robust tracking to long-term partial occlusions and intersections in terms of reducing tracking drift and tracking failure. See Paper A, for more details.

### 4.1.3 A Hybrid Method Combining Adaptive Appearance with Point Feature Tracking

The block diagram of the online learning of local and global appearance features with proposed hybrid scheme - 2 is shown in Fig.4.2. For tracker-A, feature point corre-

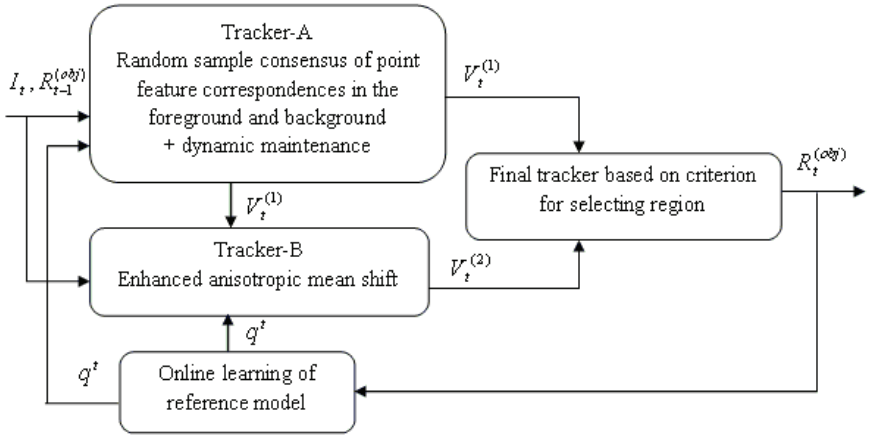


Figure 4.2: Block diagram of hybrid tracking scheme - 2

spondences are estimated through using maximum consensus point correspondences in the foreground and background areas by SIFT [13] and RANSAC [42]. It generates a parameterized candidate region $R_t^{(1)} : V_t^{(1)} = [x^{(1)}, y^{(1)}, w^{(1)}, h^{(1)}, \theta^{(1)}]_t^T$, corresponding to the 2D center, width, height and orientation of the region. Apart from a dynamic point maintenance and the use of separate foreground and background areas, a re-initialization process is applied to tracker-A if the similarity between the tracked area and the reference object area becomes small, indicating a potential tracking drift ( e.g., due to few correspondence points) which could propagate through frames.

For tracker-B, an enhanced anisotropic mean shift is achieved by choosing the center between the candidate region of tracker-A and the previous candidate region of mean shift [40], and by allowing a re-initialization process. The basic idea is to guide the MS to a correct target object location especially when confusing track situations occur (e.g. other objects with similar color distributions, or cluttered), through assigning the tracker to an area that is more agreeable with the local feature correspondences of the target. It generates a parameterized candidate region $R_t^{(2)} : V_t^{(2)} = [x^{(2)}, y^{(2)}, w^{(2)}, h^{(2)}, \theta^{(2)}]_t^T$. A 3rd candidate object region

$R_t^{(3)} : V_t^{(3)} = \sum_{i=1}^{2} \tilde{\rho}_t^{(i)} V_t^{(i)}$ where $\tilde{\rho}_t^{(i)} = \frac{\rho_t^{(i)}}{\rho_t^{(1)}+\rho_t^{(2)}}$.

The optimal target object region $R_t^{(obj)}$ from the final tracker is then selected by maximizing the following criterion, $\hat{V}_t^{(obj)} = \arg\max_{V_t^{(i)}}(\rho_t^{(1)}, \rho_t^{(2)}, \rho_t^{(3)})$ where $\rho_t^{(i)}$, $i$=1,2,3, is the Bhattacharyya coefficient measuring the similarity between the reference and the candidate object from the tracked candidate area $R_t^{(i)}$. The online learning of the reference appearance distribution $q_u^t$ is also applied. See Paper B, for more details.

## 4.2 Object Appearance Learning and Tracking on Grassmann and Symmetric Manifolds

This is achieved by: i) Computing posterior manifold point (object appearance) estimate at each new observation by Bayesian formulation; ii) Defining dynamic model that uses two state variables for modeling the online learning process on manifolds: one is for object appearances on manifolds, another is for velocity vectors in tangent planes of manifolds; iii) Measuring likelihood from the predicted manifold points and the current observation by geodesics; iv) Exploiting another Bayesian formulation for tracking object bounding box parameters by embedding the manifold appearance. From an application point of view, the proposed framework is used in Grassmann and symmetric manifolds for better tracking performance.

The rest of the section is organized as follows: section 4.2.1 summarizes the algorithmic framework for object appearance online learning on manifolds. The object tracking on manifolds is described in section 4.2.2. Finally, the methodology is exemplified with Grassmann and symmetric manifolds.

### 4.2.1 Object Appearance Learning

The proposed Bayesian formulation for online learning of object appearances on manifolds can be detailed as follows:

#### State Vector

Let the object appearance at $t$ be described by a point on a manifold and the speed of appearance change. We define the state vector as follows:

$$\mathbf{s}_t = [\mathbf{p}_t, \boldsymbol{\Delta}_t]^T \tag{4.6}$$

where $\mathbf{p}_t$ is the object appearance point on the manifold and $\boldsymbol{\Delta}_t$ is the corresponding velocity for $(\mathbf{p}_{t-1}, \mathbf{p}_t)$ with $\mathbf{p}_t$ be the end point of geodesic starting from $\mathbf{p}_{t-1}$.

#### Dynamic Model

The state (i.e., object appearance) dynamics be described by the following model:

$$\begin{cases} \mathbf{p}_t = h(\mathbf{p}_{t-1}, \boldsymbol{\Delta}_t) = \exp_{\mathbf{p}_{t-1}}(\boldsymbol{\Delta}_t) \\ \boldsymbol{\Delta}_t = \boldsymbol{\Delta}_{t-1} + \mathbf{V}_1 \end{cases} \tag{4.7}$$

where $\mathbf{V}_1$ (including the acceleration and model noise) is assumed to be zero-mean white, $\mathbf{\Delta}_{t-1}$ is constant in each sample interval $T = t_k - t_{k-1}$, and $T = 1$ is set for mathematical convenience, $h(\cdot)$ is nonlinear, $\exp_{\mathbf{p}_{t-1}}(.)$ is calculated by using exponential mapping formula of manifold under consideration. The rational behind dynamic model is to exploit the stochastic process on the manifold as a piecewise-geodesic curve with random velocities at individual pieces.

Predicted manifold points $\mathbf{p}_t^j$ can be obtained as follows: Let $\mathbf{p}_{t-1}^j$ be the previous manifold particle point at $t-1$ and $\mathbf{\Delta}_{t-1}^j$ be the velocity particle that connects $(\mathbf{p}_{t-2}^j, \mathbf{p}_{t-1}^j)$, where $\mathbf{p}_{t-1}^j$ is the end point of the geodesic starting from $\mathbf{p}_{t-2}^j$. First, a set of velocity particles $\mathbf{\Delta}_t^j$ (originated from $\mathbf{p}_{t-1}^j$) is generated in tangent planes using the previous velocity particles $\mathbf{\Delta}_{t-1}^j$ according to $\mathbf{\Delta}_t^j = \mathbf{\Delta}_{t-1}^j + \mathbf{V}_1$ (the 2nd equation of (4.7)), $j = 1, \cdots, N_1$. Then, a set of new manifold particles $\mathbf{p}_t^j$ is obtained from $\mathbf{\Delta}_t^j$ through the exponential mapping, according to $\mathbf{p}_t^j = exp_{\mathbf{p}_{t-1}^j}(\mathbf{\Delta}_t^j)$ (the 1st equation of (4.7)). These manifold points $\mathbf{C}_t^j$ are considered as the predicted points at $t$.

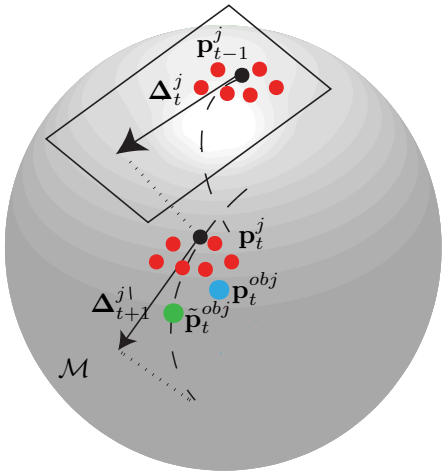


Figure 4.3: Relationship of manifold points and tangent vectors used in the learning process. From $t-1$ to $t$: red dots on the top cluster are manifold particles $\mathbf{p}_{t-1}^j$ at $t-1$. Consider one manifold particle $\mathbf{p}_{t-1}^j$ (black dot) and its velocity particle (black arrow) $\mathbf{\Delta}_t^j$ in its tangent plane. Mapping this vector through the geodesic (dash curve) on the manifold results in a predicted particle at t. Repeat this process to all manifold particles leads to a set of predicted particles (red dots in the bottom cluster). The posterior $\mathbf{p}_t^{obj}$ (blue dot) at $t$ is then obtained from the predicted particles and the new observation $\tilde{\mathbf{p}}_t^{obj}$ (green dot).

**Likelihood**

It is modeled as the Gaussian distribution of distance $d(\mathbf{p}_t^{obj}, \mathbf{p}_t^j)$ between the current observation $\mathbf{p}_t^{obj}$ and predicted particles $\mathbf{p}_t^j$ as:

$$p(\tilde{\mathbf{p}}_t^{obj}, \mathbf{p}_t^j) = \exp\left\{-\frac{d(\tilde{\mathbf{p}}_t^{obj}, \mathbf{p}_t^j)^2}{\sigma_l^2}\right\} \tag{4.8}$$

where $\sigma_l^2$ is the measurement noise ($\sigma_l^2 = .1$ in our tests), $\|.\|$ is calculated by using the distance formula of manifold under consideration. The current observation $\tilde{\mathbf{p}}_t^{obj}$ at time instant $t$ is provided by the tracking process (see Section 4.2.2).

The weight is then updated as follows:

$$w_t^j \propto w_{t-1}^j p(\tilde{\mathbf{p}}_t^{obj}|\mathbf{p}_t^j) \tag{4.9}$$

and subsequently normalized by $w_t^j = w_t^j / \sum_j w_t^j$. Resampling is applied if $\hat{N}_{eff} = 1/\sum_j (w_t^j)^2 < N1_{th}$, to prevent the degeneracy problem [76].

**Posterior Online Learned Manifold Point**

Finally, the MMSE estimate of the object appearance manifold point $\mathbf{p}_t^{obj}$ is obtained by the expected value of weighted predicted particles on the manifold. The expression for expectation is dependent on the choice of manifold (See paper C and D for more details).

Fig.4.3 shows the relationship of manifold particles, velocity particles, observation point, and the predicted and posterior estimated points between two time instants.

### 4.2.2 Object Tracking

The aim in the tracking process is to find the MAP (maximum a posteriori) estimate of the object bounding box (and the image within the bounding box). This is realized by utilizing another PF. Let the state vector $\mathbf{s}2_t = [y_t^1 \ y_t^2 \ \beta_t \ \gamma_t \ \alpha_t \ \phi_t]^T$ at $t$ be the affine bounding box parameters (2D box center, scale, rotation, aspect ratio and skew). Given a set of particles at $t-1$, a new set of particles $\{\mathbf{s}2_t^i\}_{i=1}^{N_2}$ is generated by PF-2 according to the state equation $\mathbf{s}2_t = \mathbf{s}2_{t-1} + \mathbf{v}2$ where $\mathbf{v}2 \sim \mathcal{N}(0, \mathbf{\Omega})$. The likelihood is modelled as Gaussian distributed distance of dynamic prediction error of particles on the manifold. The expression for the likelihood is dependent on the manifold under consideration. Finally, object bounding box MAP (maximum a posteriori) estimate is computed from posterior pdf. Using the estimated $\mathbf{s}2_t^{obj}$, the corresponding appearance $\tilde{\mathbf{p}}_t^{obj}$ to the object within the box is computed and provided to the online learning process at $t$.

### 4.2.3 Application to Grassmann Manifolds

Fig.4.4 shows the block diagram of the proposed scheme. The proposed scheme can be split into two parts: the block-1 on the top is for a Bayesian object tracking process,

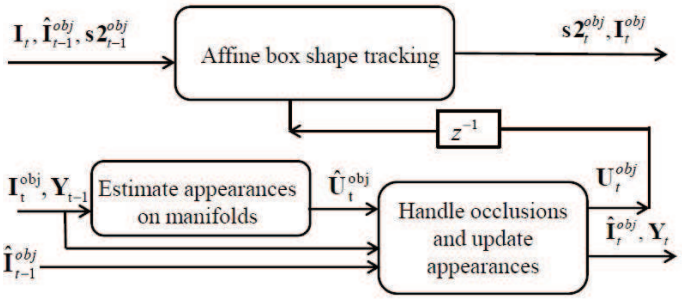


Figure 4.4: Block diagram of the proposed integrated scheme. $\hat{\mathbf{I}}_{t-1}^{obj}$ is the reference object image at t-1, $\hat{\mathbf{I}}_t^{obj}$ and $\mathbf{s2}_{t-1}^{obj}$ are the tracked object image and its box parameters at t-1; $\mathbf{I}_{t-1}^{obj}$ is tracked object image used as the new observation image at t; $\hat{\mathbf{U}}_t$ is the estimated manifold appearance, and $\mathbf{U}_t^{obj}$ is the final updated appearance after occlusion handling; ($\mathbf{Y}_{t-1}$, $\mathbf{Y}_t$) are the observation matrix with a sliding window size L at t-1 and t; $\mathbf{I}_t$ is the current video frame; and $z^{-1}(\mathbf{U}_t^{obj}) = \mathbf{U}_{t-1}^{obj}$ is the reference object appearance at t-1 used for the tracking process at t.

and the block-2 (bottom left) and block-3 (bottom right) are for process of Bayesian manifold online appearance estimation, and manifold updating with criterion-based occlusion handling, respectively. In the tracking process, object bounding box affine parameters are tracked by a particle filter. In the online updating process, the appearance subspace is first estimated on the Grassmann manifold by another particle filter. The particle filter uses a nonlinear dynamic model, the exponential and logarithmic mapping functions between the tangent planes and the manifold. The likelihood in this particle filter is computed by the subspace angles between the current observation and predicted manifold particles. The online learned object appearance is then obtained as the posteriori manifold point. A criterion is applied to estimate the occlusion. If no occlusion is detected, updating the basis matrix of reference object appearance is then performed. These two parts, tracking and updating, are performed in an alternation fashion as an integrated tracking scheme. See Paper D, for more details.

### 4.2.4 Application to Symmetric Manifolds

Fig.4.5 shows the complete tracking scheme for symmetric manifold. In Block 1 (top block) particle filter-1 is utilized for updating the object appearance covariance matrix on the symmetric manifold by using the tracked object appearance and the predicted manifold points from the dynamic appearance model. The dynamic model is realized by computing the velocity vector of candidate manifold points under a constant velocity assumption and then mapping the velocity vector to the manifold originated from the previous manifold point. In Block 2 (middle block), the affine shape parameters of objects bounding box is tracked by particle filter-2. In Block

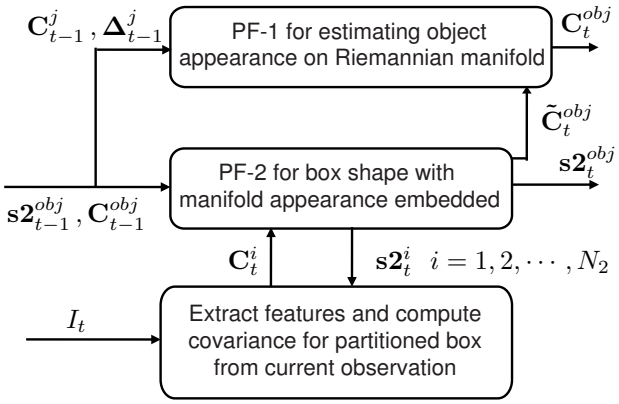


Figure 4.5: Block diagram of the proposed integrated scheme. ($\mathbf{C}_{t-1}^{obj}$, $\mathbf{C}_{t}^{obj}$), ($\mathbf{s}2_{t-1}^{obj}$ and $\mathbf{s}2_{t}^{obj}$) are the manifold appearance and box parameter vector of tracked object at t-1 and t; $\mathbf{C}_{t}^{i}$ and $\mathbf{s}2_{t}^{i}$ are the manifold appearance and box parameter vector of $i$th candidate object at t; $\mathbf{C}_{t-1}^{j}$ and $\boldsymbol{\Delta}_{t-1}^{j}$ are the manifold appearance and velocity vector of candidate object at t-1; $\tilde{\mathbf{C}}_{t}^{obj}$ is the new observation yielded from the tracking process.

3, the covariance matrix of object appearance is computed from Gabor features in partitioned regions. See Paper C, for more details.

CHAPTER 5

# Conclusion and Future Work

In this thesis, four new algorithms, are proposed for visual object tracking and online learning. Among them, two algorithms operate on the Euclidean space while the other two take into account for the manifold nature of visual data. Each of the proposed algorithms has limitations, advantages and disadvantages. This knowledge can be used for the choice of a particular algorithm.

For joint anisotropic mean-shift and particle-filter framework, employing online learning, fully adjustable bounding box, multiple mode appearance modeling and efficient re-distribution of particles have all contributed to the performance improvement of the proposed tracker. The required number of particles in the particle filter is very small (15 in our tests). Compared to existing techniques, improved results are obtained for video scenes containing similar objects, frequent intersections and partial occlusions. However, several empirically determined parameters pose a limitation to the general use of the proposed scheme. It is further noticed that the proposed scheme is somewhat sensitive to pose changes of large objects, e.g. a large bounding box only contains a face. In such a scenario, the bounding box would sometimes become non-tight, or deviate from the object center (e.g. part of the object is outside the box).

For joint point feature correspondences and object appearance similarity, introducing dynamic maintenance to foreground and background feature point sets, enhancing the mean shift by exploiting local features as the guide and dynamic updating the reference object model, as well as utilizing an optimal selection criterion for the final tracking have shown to improve the tracking results. However, It is observed that if a target object in the video experiences long-duration partial occlusions over a large percentage of area (e.g.>60%) then the tracking performance can be degraded, especially if the visible part is rather smooth and lacks local feature points. In principle, full or partial occlusions of long duration is beyond the limitation of this scheme.

Gabor features on partitioned object bounding boxes are shown to be effective and very robust for tracking visual and infrared objects. Symmetric manifold online

learning and tracking scheme at each time instant is effective. It leads to better tracking results especially for objects with significant non-planar pose changes from videos made from a single camera. However, Online learning relating to object occlusions is not dealt with in this study.

For Grassmann manifold, the online learning of appearance subspace and box shape tracking by integrating dynamic appearance and shape on the manifold is shown to be efficient. This is a better framework for tracking as it allows motion models and appearance models to be accurately described. Moreover, a method to detect partial occlusion is shown to be effective. However, It is observed that if a target object in the video, undergoes large pose change with long-duration partial occlusions then the tracking performance can be degraded.

## 5.1 Future Work

In practical visual tracking systems, tight bounding box, ability to handle occlusion, low drift rate are desirable factors and it accounts as a measure of tracker stability and robustness. The proposed approach have shown some promising results in achieving these objectives, but certainly is not perfect. Several improvements need to be addressed in the near future research.

### 5.1.1 Adaptive Selection of Emperical Parameters

Adaptive mechanism for the emperical parameter of proposed methods may result in more generic and robust tracker and at the same time avoids manual case-by-case adjustments.

### 5.1.2 Real Time Applications

Computational time for the proposed tracking scheme still poses a problem and limits the real-time applications. To overcome this limitation, C-programs with optimized codes, or FPGA can be considered to replace the Matlab programs used in current tests.

### 5.1.3 Integration of Multiple Visual and Infrared Cameras

Another possible research direction, towards building robust trackers, can be the integration of multiple visual and infrared cameras. It may increase tracker ability against partial occlusion of large areas or even full occlusions and day/night capability. This is the most desirable feature for surveillance applications.

### 5.1.4 Objects Detection and Activity Analysis

Detection of interesting objects and analysis of object tracks to recognize their behavior, before and after visual tracking, can be one more possible research direction. Integration of these modules can make the trackers very adaptive to users and robust.

# Bibliography


[1] D. Comaniciu, V. Ramesh and P. Meer, "Kernel-based object tracking", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 5, pp. 564-577, 2003.

[2] A.Yilmaz, O.Javed and M.Shah, "Object tracking: A survey", ACM Comput. Surv., vol. 38, no. 4, 2006.

[3] C. J. Veenman, M. J. T. Reinders, and E. Backer, "Resolving motion correspondence for densely moving points", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 23, no. 1, pp. 54-72, 2001.

[4] K. Shafique and M. Shah, "A noniterative greedy algorithm for multiframe point correspondence", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 27, no. 1, pp. 51-65, 2005.

[5] M. J. Black and A. D. Jepson, "EigenTracking: robust matching and tracking of articulated objects using a view-based representation ", Int. Journal of Computer Vision, vol. 26 no. 1, pp. 63-84, 1998.

[6] I. Haritaoglu, D. Harwood and L. S. Davis, "W4: real-time surveillance of people and their activities ", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 22 no. 8, pp. 809-830, 2000.

[7] A. Ali and J. Aggarwal, "Segmentation and recognition of continuous human activity", in Proc. IEEE Workshop on Detections and Recognition of Events in Video, pp. 28-35, 2001.

[8] N. Paragios and R. Deriche, "Geodesic active contours and level sets for the detection and tracking of moving objects", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 22, no. 3, pp. 266-280, 2000.

[9] B. Lucas and T. Kanade, "An iterative image registeration technique with an application to stereo vision ", in proc. Int. Joint Conf. on Artifical Intellegence , pp. 674-679, 1981.

[10] O.Tuzel, F. Porikli and P. Meer, "Region covariance: a fast descriptor for detection and classification", Proc. ECCV, pp. 589-600, 2006.

[11] C. Harris and M. Stephens, "A Combined Corner and Edge Detector", in Proc. The Fourth Alvey Vision Conf., pp. 147-151, 1988.

[12] C. Tomasi and T. Kanade, "Shape and motion from image streams: A factorization method part3 - detection and tracking of point features", Technical report CMU-CS-TR, 1991.

[13] D. G. Lowe, "Distinctive Image features from scale-invariant keypoints", Int. Journal of Computer Vision, vol. 60, pp. 91-110, 2004.

[14] H. Bay, T. Tuytelaars, L. V. Gool, "SURF: Speeded Up Robust Features", in proc. ECCV, LNCS 3951, pp. 404-417, 2006.

[15] G. D. Hager and P. N. Belhumeur, "Real-time tracking of image regions with changes in geometry and illumination", Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition, pp. 403-410, 1996.

[16] J. Shi and C. Tomasi, "Good features to track", in Proc. IEEE Computer Society Conf. Comp. Vision and Pattern Recognition, pp. 593-600, 1994.

[17] D. Comaniciu and P. Meer, "Mean shift analysis and applications ", Proc. IEEE Int. Conf. Computer Vision, vol. 2, pp. 1197-1203, 1999.

[18] J. Shi and J. Malik, "Normalized cuts and image segmentation", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 22 no. 8, pp. 888-905, 2000.

[19] C. Stauffer and W. E. L. Grimson, "Learning patterns of activity using real-time tracking", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 22 no. 8, pp. 747-757, 2000.

[20] N. M. Oliver, B. Rosario and A. P. Pentland, "A Bayesian computer vision system for modeling human interactions", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 22 no. 8, pp. 831-843, 2000.

[21] A. Monnet, A. Mittal, N. Paragios and V. Ramesh, "Background modeling and subtraction of dynamic scenes", Proc. IEEE Int. Conf. Computer Vision, pp. 1305-1312, 2003.

[22] C. P. Papageorgiou, M. Oren, T. Poggio, "A general framework for object detection", Proc. IEEE Int. Conf. Computer Vision, pp. 555-562, 1998.

[23] K. Shafique and M. Shah, "Neural Network-Based Face Detection", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 20, no. 1, pp. 23-88, 1998.

[24] P. Viola, M. J. Jones and D. Snow, "Detecting pedestrians using patterns of motion and appearance", Proc. IEEE Int. Conf. Computer Vision, pp. 734-741, 2003.

[25] Y. B-Shalom, X. R. Li and T. Kirubarajan "Estimation with applications to tracking and navigation theory algorithms and software", Willey-Interscience Publications, 2001.

[26] P. S. Maybeck Chen, "Stochastics models, estimation and control", vol. 2, New York: Academic Press, 1982.

[27] R. Rosales and S. Sclaroff, "3D trajectory recovery for tracking multiple objects and trajectory guided recognition of actions ", in Proc. IEEE Computer Society Conf. Comp. Vision and Pattern Recognition, pp. 117-123, 1999.

[64] R. V. D. Merwe, A. Doucet, N. D. Freitas and E. Wan, "The unscented particle filter", Technical report CUED/F-INFENG/TR 380, Cambridge University Engineering Department, 2000.

[29] B. Ristic, S. Arulampalam and N. Gordon "Beyond the Kalman Filter: Particle Filters for Tracking Applications", Artech House Publishers, 2004.

[30] A. Doucet, N. D. Freitas and N. Gordon, "Sequential Monte Carlo Methods in Practice", New York: Springer, 2001.

[31] A.C.Sankaranarayanan, A. Veeraraghavan, R.Chellappa, "Object Detection, Tracking and Recognition for Multiple Smart Cameras",Proceedings of the IEEE, Vol.96, No.10, pp. 1606-1624, 2008.

[32] G. Hager and P. Belhumeur, "Real-time tracking of image regions with changes in geometry and illumination", Proc. CVPR, 1996.

[33] Q. Zhao, S. Brennan and H. Tao, "Differential EMD tracking", In PProc. ICCV, 2007.

[34] D. Ross, J. Limy, R. Line and M. Yang, "Incremental learning for visual tracking", IJCV, 2007.

[35] S. Avidan, "Ensemble tracking", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 29, pp. 561-271, 2007.

[36] H.Seung, D.Lee, "The manifold ways of perception", Sci2ence, 290(5500), pp.2268-2269, 2000.

[37] A.Edelman, T.A.Arias, S.T,Smith, "The geometry of algorithms with orthogonality constraints", SIAM J. Matrix Anal. Appl., 20(2), 1998.

[38] A.Srivvastava, E.Klassen, "Bayesian and geometric subspace tracking", Adv. Appl. Prob. (SGSA), vol.36, pp.43-56, 2004.

[39] T.Wang, A.G.Backhouse,I.Y-H.Gu, "Online subspace learning on Grassmann manifold for moving object tracking in video", Proc. ICASSP , 2008.

[40] Q. Sumin and H. Xianwu, "Hand tracking and gesture gecogniton by anisotropic kernel mean shift", in Proc. IEEE Int. Conf. Neural Networks and Signal processing, vol. 25, pp. 581-585, 2008.

[41] C. Shan, Y. Wei, T. Tan and F. Ojardias, "Real time hand tracking by combining particle filtering and mean shift", in Proc. IEEE Int. Conf. Automatic Face and Gesture Recognition, pp.669-674, 2004.

[42] M.A.Fischler and R.C.Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartographys", Communications of the ACM, vol. 24, pp. 381-395, 1981.

[43] D.Ross, J.Lim, R.S.Lin, M.H.Yang, "incremental learning for robust visual tracking", Int. J. Comput. Vis., 77(1), pp.125-141, 2008.

[44] F.Porikli, O.Tuzel, P.Meer, "Covariance tracking using model update based on Lie algebra", Proc. IEEE CVPR conf., pp.728-735, 2006.

[45] Y.Wu, B.Wu, J.Liu, H. Lu, "Probabilistic tracking on Riemannian manifolds", Proc. ICPR conf., 4 pages, 2008.

[46] R.Subbarao, P.Meer, "Nonlinear mean shift over Riemannian manifolds", Int. J. Comput. Vis., 84(1), pp.1-20, 2009.

[47] M.F. Abdelkader, W.A-. Almageed, A. Srivastava, R. Chellapa "Silhouette-based gesture and action recognition via modeling trajectories on Riemannian shape manifolds", Journal Computer Vision and Image Understanding, vol. 115, no. 3, pp. 439-455 , 2011.

[48] J. Gallier, "Notes on diffrential geometry and Lie groups", Department of computer and information science, University of PPennsylvania, USA, 2010.

[49] P.A.Absil, R. Mahony and R.Sepulchre, "Optimization algorithms on matrix manifolds", Princeton University Press, ISBN: 978-0-691-13298-3, 2008.

[50] J.M.Lee, "Introduction to smooth manifolds", Springer, ISBN-13: 978-0387-9448-6, 2006.

[51] J. H. Manton, "On the role of differential gepmetry in signal processing", ICASP 2005, 2005.

[52] B.O'Neill, "Semi-Riemannian manifold: with applications to relativity", Pure and applied mathematics, ISBN 9780125267403, Volume 103, 468, 1983.

[53] E. Begelfor and W. Werman, "Affine invariance revisted", Proc. IEEE CVPR, pp. 2087-2094 , 2006.

[54] K. A. Gallivan, A. Srivastave, X. Liu, "Efficient algorithms for inferences on Grassmann manifolds", Proc. IEEE workshop on Statistical Signal Processing, pp.315-318, 2003.

[55] A. Brun, "Manifolds in image science and visualization", Ph.D. Thesis, Department of Biomedical Engineering, Linköping University, Sweden, 2007.

[56] X.Pennec, P.Fillard, N. Ayache, "A riemannian framework for tensor computing", Int.J. Comput. Vision, 66(1), pp.41-66, 2006

[57] V.Arsigny, P.Fillard, X.Pennec, N.Ayache, "Geometric means in a novel vector space structure on symmetric-positive definite matrices", SIAM J. Matrix Anal. Appl., 66(1), pp.328-347, 2008.

[58] Y. B. Shalom, "Multitarget-multisensor tracking: Principles and techniques", YBS publishing, 1995.

[59] S. Blackman and R. Popoli "Design and analysis of modern tracking systems", Artech House Publishers, 1999.

[60] N. J. Gordon, A. Doucet and N. D. Freitas, "On sequential monte carlo sampling methods for bayesian filtering", Statistics and Computing, vol. 10, pp. 197-208, 2000.

[61] N. J. Gordon, D. J. Salmond and A. F. M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation", IEE Proceedings-F Radar and Signal Processing, vol. 140, pp. 107-113, 1993.

[62] M. Pitt and N. Shephard, "Filtering via simulation: Auxilary particle filters", Journal of the American Statistical Association , vol. 94, no. 446, pp. 590-599, 1999.

[63] C. Musso, N. Oudjane and F. LeGland, "Improving redularised particle filters", Sequential Monte Carlo Methods in Practice, New York: Springer, 2011.

[64] R. van der Merwe, A. Doucet, N. de Freitas and E. WanN, "The Unscented Particle Filter", Tech Rep. CUED/F-INFENG/TR 380, Cambridge University Engineering Department, 2000.

[65] G. Kitagawa, "Monte Carlo filter and smoother for non-Gaussian non-linear state space models", Journal of Computational and Graphical Statistics, vol. 5, no. 1, pp. 1-25, 1996.

[66] J. S. Liu and R. Chen, "Sequential Monte Carlo methods for dynamical systems", Journal of the American Statistical Association, vol. 93, pp. 1032-1044, 1998.

[67] K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function with applications in pattern recognition", IEEE Trans. Information Theory, vol. 21, pp. 32-40, 1975.

[68] Y. Cheng, "Mean shift, mode seeking and clustering", IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 17, no. 8, pp. 790-799, 1995.

[69] J. Wang,ăB. Thiesson,ăY. Xu,ăM. F. Cohen, "Image and Video Segmentation by Anisotropic Kernel Mean Shift", Proc. ECCV (2)'2004, pp. 238-249, 2004.

[70] I. Y. H. Gu, V. Gui, "Joint Space-Time-Range Mean Shift-based Image and Video Segmentation (Chapter 6)", Advances in Image and Video Segmentation (edited by Y-J.Zhang), Idea Group Inc. Publishing, pp. 113-139, ISBN/ISSN: 1-59140-753-2, 2006.

[71] L. Bretzner and T. Lindeberg, "Feature Tracking with Automatic Selection of Spatial Scales", in Proc. Comp. Vision and Image Understanding, vol. 71, pp. 385-392, 1998.

[72] N. Snavely, "Computer Vision CS6670", Department of Computer Science, Cornell University, Ithaca, NY,2009

[73] C.O.Tuzel, "Learning on Riemannian manifolds for interpretatio of visual enviornments", Ph.D. thesis New Brunswick Rutgers, The state university of New Jersey, 2008.

[74] R. Subbarao, "Robust statistics over Riemannian manifolds for computer vision", Ph.D. thesis New Brunswick Rutgers, The state university of New Jersey, 2008.

[75] Petter Strandmark, Irene Y.H. Gu, "Joint Random Sample Consensus and Multiple Motion Models for Robust Video Tracking", in Springer LNCS Vol. 5575, (for 16th Scandinavian Conference on Image Analysis, Oslo, Norway, July 15-18, 2009), pp. 450-459, 2009.

[76] A.Dore, M.Soto, C.Regazzoni, "Bayesian tracking for video analytics", IEEE Trans. Image Processing, Vol.27, No.5, pp.46-55, 2010.

[77] D. A. Forsyth and J. Ponce, "Computer vision: A modern approach", Prentice Hall, August 2002.

[78] S.Haner, I.Y.H. Gu, "Combining foreground/background feature points and anisotropic mean shift for enhanced visual object tracking", International Conference on Pattern Recognition (ICPR 2010), pp.3488-3491, 2010.